Relatório do Trabalho Final de Curso

Implementação da Instrumentação de um Helicóptero Autónomo

Raul Morgado 0553
Tiago Costa 0562

Lisboa, Novembro de 2002

Instituto Superior Técnico
Departamento de Engenharia Electrotécnica e de Computadores
Trabalho Final de Curso 573/L, 2001/2002

Trabalho coordenado por:
Prof. Carlos Silvestre
Prof. Paulo Oliveira

Agradecimentos

O trabalho final de curso, que usualmente se apresenta ao aluno como a última meta a alcançar, possibilita a aprendizagem de novos temas, o que torna um projecto desta dimensão bastante aliciante. Desta forma, queremos agradecer a todos aqueles que tornaram a nossa passagem pelo Instituto Superior Técnico uma mais valia para a vida futura que teremos pela frente.

Como tal agradecemos ao professor responsável pelo trabalho, Prof. Carlos Silvestre, por todo o apoio e encorajamento prestado e principalmente por nos ter auxiliado numa hora difícil e decisiva, ao nos receber neste seu projecto.

Agradecemos ao Prof. Paulo Oliveira por sempre se mostrar prestável em qualquer altura e pelo seu vasto conhecimento que se mostrou numa ajuda preciosa na solução de alguns problemas que se afiguravam mais difíceis.

Agradecemos ao Eng.º Manuel Rufino, ao Eng.º João Alves e ao Eng.º Luís Sebastião, por terem prestado um auxílio importante e permitido um melhor desenvolvimento do trabalho.

Agradecemos ao Sr. João Serralha pela sua boa disposição e pela forma como todos os dias mantêm o laboratório de trabalho em perfeitas condições para o aluno poder desenrolar o seu trabalho, sem que nada o prejudique.

Por fim um agradecimento especial à nossa família e amigos que tantas vezes se viram privados da nossa companhia em momentos que deviam ter sido deles, e mesmo assim se mostravam prestáveis para nos ajudar e apoiar.

Resumo

Tendo como base de suporte outros projectos da mesma linha de raciocínio do Instituto de Sistemas e Robótica é objectivo deste trabalho final de curso a Implementação da Instrumentação de um Helicóptero Autónomo. Para isso foi desenvolvida inicialmente uma pesquisa acerca do assunto de modo a ter-se uma ampla visão do problema proposto.

A Instrumentação de um helicóptero Autónomo visa a possibilidade de controlar um pequeno aparelho na execução de tarefas que se apresentam de elevado risco, em termos de segurança, ou mesmo impossíveis de executar por um operador humano. Para a realização deste objectivo teve-se de elaborar um circuito (placa de amostragem de sinais) que permita ao operador humano ter um controlo quase absoluto sobre a unidade móvel (helicóptero), isto é, receber leituras correctas relativas à posição, velocidade, altitude e atitude da unidade referida. Para isso o circuito encontra-se em comunicação com um conjunto de sensores que fornecem os dados referentes a cada uma das grandezas, tendo o circuito a função de conversão dos dados para formato digital e fornece-los com o maior grau de precisão possível ao operador.

Indíce

Lista de Figuras

Lista de Tabelas

Lista de Siglas

ADC	*Analog-to-Digital Converter*
OSR	*Oversampling Ratio*
PCI	Placa de Circuito Impresso
SNR	*Signal-to-Noise Ratio*
ENOB	*Effective number of bits*
PAS	Placa de Aquisição de Sinais
PGA	*Programmable Gain Amplifier*
TM	*Turbo Mode*
INSR	*Instruction Register*
DOR	*Data Output Register*
OCR	*Offset Calibration Register*
FCR	*Full-Scale Calibration Register*
FFT	*Fast Fourier Transform*

1 Introdução

1.1 Motivação

A Instrumentação de um Helicóptero Autónomo apresenta-se não só como um projecto novo, como também, e essencialmente, como um desafio a todos os conhecimentos adquiridos enquanto estudantes. Os principais factores que influenciaram a decisão da escolha deste trabalho foi o facto deste se apresentar bem estruturado, ter uma forte componente prática, ser um projecto para possível aplicação, envolvendo uma das aeronaves existentes mais complexas e instáveis, e de possível utilização em diversas tarefa aéreas que lhe sejam destinadas, num ambiente não estruturado e de difícil acesso por outros meios, sem a intervenção de um operador humano.

1.2 Objectivos

O trabalho tem como principal objectivo a aquisição de sinais provenientes de sensores (magnetómetros, giroscópios, acelerómetros) que medem o estado do helicóptero, e que posteriormente serão convertidos e utilizados na estabilização da plataforma, permitindo assim um elevado grau de autonomia do mesmo. Para isso foi necessário o desenvolvimento de uma Placa de Circuito Impresso (PCI) e de uma arquitectura de *software,* possibilitando uma fácil aquisição de dados de saída pelo utilizador.

Um dos parâmetros de qualidade a atingir será uma resolução efectiva superior a sistemas de aquisição de dados existentes no mercado, ou seja, obter leituras o mais fiáveis possíveis tendo em consideração as características dos diversos componentes utilizados.

1.3 Enquadramento

Este trabalho final de curso insere-se no seguimento de outros dois trabalhos realizado por alunos no ano transacto, e visa complementar e melhorar esses mesmos trabalhos.

Neste trabalho pretende-se atingir um elevado grau de precisão dos dados referentes ao actual estado do helicóptero, e substancialmente melhor do que a conseguida anteriormente.

No fim deste projecto, espera-se ter atingido mais uma etapa na direcção do objectivo final, a realização de uma forma eficiente de tarefas difíceis a um ser humano, ou que o poderão colocar em risco, por meio de um helicóptero.

1.4 Estrutura do Relatório

Este relatório de trabalho final de curso encontra-se dividido em cinco capítulos.

No presente capítulo começa-se por dar a conhecer o âmbito do projecto e sua descrição.

No capítulo 2, apresenta-se a arquitectura da PCI desenvolvida, nomeadamente, a arquitectura de controlo utilizada e descrição do Conversor Analógico-Digital, justificando a sua escolha.

No capítulo 3, apresenta-se a arquitectura de aquisição de sinais e suas especificações (parte analógica e parte digital).

No capítulo 4, descreve-se a arquitectura de controlo da temperatura dos sensores, sua estrutura e funcionalidades.

No quinto e último capítulo do relatório do trabalho final de curso são apresentadas as conclusões referentes ao mesmo.

No fim do relatório são apresentados alguns anexos que incluem o manual da placa de amostragem de sinais, o manual do *driver* e o circuito de lógica programável.

2 Desenvolvimento do Sistema/Hardware

2.1 Introdução

Tendo como final a atingir os objectivos referidos no capitulo anterior foi necessário a construção de uma placa de circuito impresso (PCI). Existem diversos factores que influenciam a tomada de decisão para proceder à escolha dos componentes electrónicos a utilizar. Como será de fácil compreensão num helicóptero autónomo o espaço disponível e a sua autonomia são restrições e consequentemente ter-se-á de utilizar componentes de reduzidas dimensões e de baixo consumo de energia. Todos estes factores fizeram com que se tivesse como principais limitações na construção da PCI os três princípios de seguida enunciados, apresentando-se de uma forma hierarquizada em relação ao grau de importância:

- Qualidade;
- Tamanho;
- Peso;

Pretende-se assim uma PCI com o mínimo de ruído electrónico possível, que seja a mais pequena e leve possível.

2.2 Tipos de Arquitecturas

De seguida apresenta-se um breve estudo das arquitecturas dos conversores analógicos-digitais de aplicação mais comum em situações práticas.

2.2.1 Aproximações Sucessivas

A técnica das aproximações sucessivas é a mais comum em conversores A/D de média velocidade, e é muito utilizada em aplicações que requerem resoluções da ordem dos 8 a 16 bits.

Esta arquitectura é baseada em aproximações do sinal de entrada a um código binário, verificando sucessivamente a aproximação para cada bit de código, até encontrar a melhor aproximação. Em cada etapa do processo é armazenado no registo de aproximações sucessivas, o valor binário.

Devido ao método de conversão requerer um ciclo de relógio para produzir um bit de resolução de dados de saída, o tempo de conversão é inversamente proporcional à resolução.
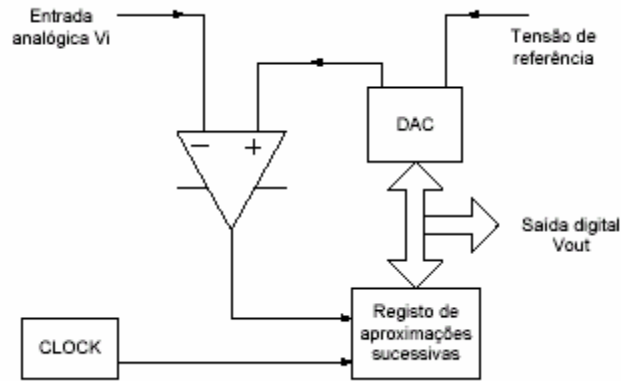
Figura 2.1 – Diagrama de Blocos da Arquitectura Aproximações Sucessivas

2.2.2 Paralelo ou Flash

O funcionamento desta arquitectura requer um divisor de tensão formado por resistências iguais, polarizado por uma ou duas referências de tensão, que definem o domínio de valores da entrada. Quando se usa uma só referência, o domínio de valores possíveis da entrada é unipolar.

A tensão de entrada é fornecida a todos os comparadores, que por sua vez estão ligados a um bloco lógico encarregado de codificar o resultado das saídas dos comparadores num número binário.

Os conversores paralelos (Flash) são muito rápidos, devido a determinarem o valor dos bits em modo paralelo.

Neste tipo de arquitecturas não é possível obter resoluções superiores a 12 bits, devido ao elevado número de comparadores e resistências que é necessário integrar num único circuito integrado.
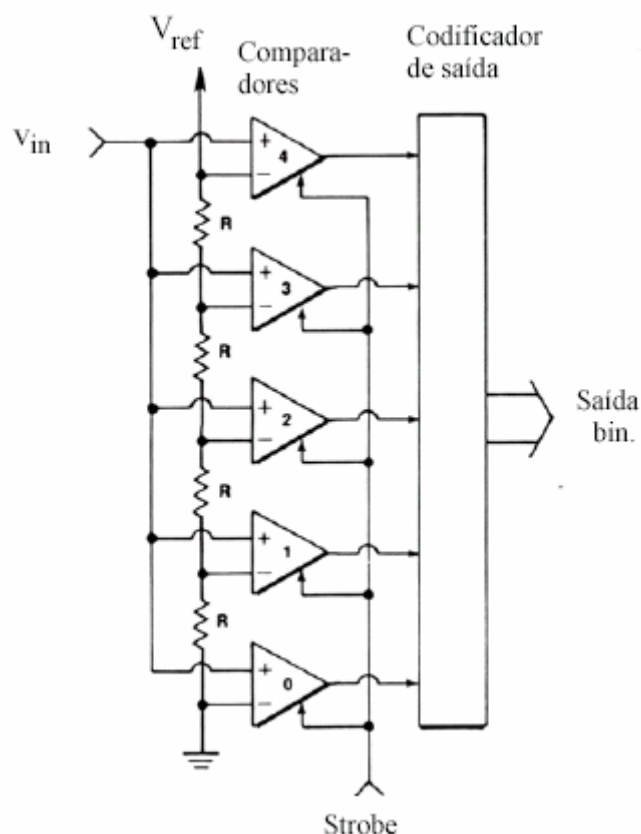
Figura 2.2 – Diagrama de Blocos da Arquitectura Flash

2.2.3 Delta-Sigma

Os ADC com modulação Delta-Sigma (ΔΣ) usam a técnica de sobreamostragem e filtragem digital para conseguir actualmente resoluções da ordem dos 24 bits.

Neste ADC o sinal analógico de entrada passa pelo modulador ΔΣ, onde é amostrado, originando apenas um bit com modulação de impulsos codificados (PCM) na sua saída. Na fase seguinte, um filtro digital passa-baixo remove o ruído de quantização a altas frequências introduzido pelo modulador e extrai uma sequência binária de N bits.

O resultado final do ADC é uma representação digital de grande aproximação do sinal analógico de entrada.

Estes conversores são especialmente usados quando é necessário uma grande resolução de sinais de baixa frequência, assim como baixa distorção de conversão. Este tipo de ADC possui uma boa linearidade e uma grande precisão.
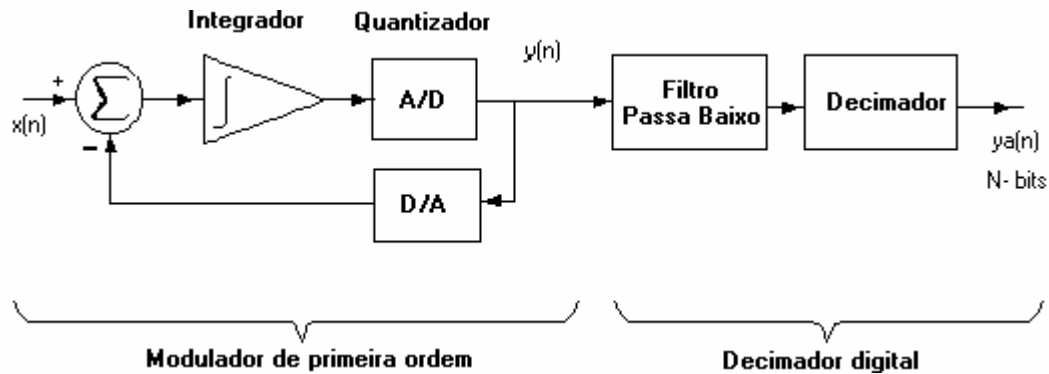
Figura 2.3 – Diagrama de Blocos da Arquitectura Delta-Sigma

2.3 Comparação das Arquitecturas

O gráfico seguinte esquematiza a comparação das arquitecturas estudadas em termos de resolução e largura de banda.
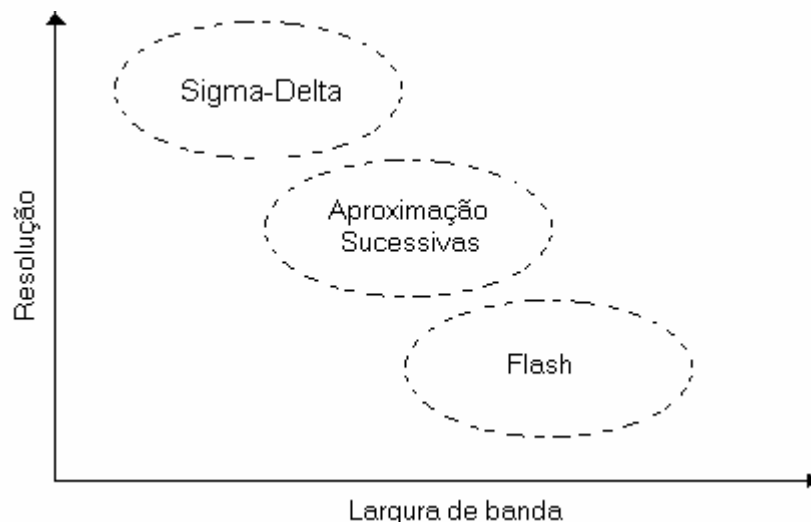
Figura 2.4 – Comparação entre Arquitecturas

Na instrumentação de um helicóptero autónomo a qualidade e a precisão dos dados em tempo real são de extrema importância. Assim sendo a arquitectura que melhor satisfaz todos os requisitos pretendidos, é a arquitectura Delta-Sigma. Esta arquitectura pode ter uma resolução até um máximo de 24 bits, tempos de conversão compatíveis com a aplicação, grande estabilidade e baixo custo.

2.4 Descrição da Arquitectura Delta-Sigma

A arquitectura dos ADC Delta-Sigma é baseada apenas num único bit por comparação, o quantizador tem apenas dois níveis de quantização e como se verá mais adiante, tornar-se-á mais atractivo em termos de desempenho que as restantes

arquitecturas. Este bit ADC é usado conjuntamente com um ritmo de sobreamostragem alto, seguido de um filtro digital. Desta forma consegue-se uma resolução de 24 bits.

Uma maneira de se entender melhor o bit ADC é pensando nele como um bit de polaridade, indicando se o sinal aumentou ou diminui relativamente à última amostra. O modulador funciona de modo que o sinal de realimentação possa seguir o sinal de entrada. A diferença dos dois sinais é continuamente comparada com um valor limiar, originando na sua saída o bit ADC, que será 1 se a diferença dos dois sinais é positiva ou 0 em caso contrário.

Os conversores Delta-Sigma são constituídos por uma parte analógica muito simples e outra digital, mas poder-se-á dizer que são essencialmente digitais, de tal modo que resulta num fabrico de baixo custo e consegue-se obter uma grande estabilidade. Apesar destes ADC terem muitos aspectos vantajosos, também tem as suas desvantagens. O caso mais negativo será obviamente a limitação da resposta em frequência a alguns KHz.

Figura 2.5 – Diagrama de Blocos de um ADC Delta-Sigma

Este tipo de ADC explora todos os benefícios da sobreamostragem, como se irá verificar. Mas o que se entende por sobreamostragem? Não é nada mais do que poder-se amostrar o sinal de entrada a um ritmo superior do ritmo mínimo, ou seja do ritmo de Nyquist.

O melhoramento da relação sinal-ruído (SNR) é conseguido por um espalhamento da potência do ruído de quantização por uma gama de frequências maior. Contudo esse melhoramento nunca vai além de 3dB por cada duplicação do ritmo de amostragem.

Considere-se uma amostra de um sinal x(t) amostrada no instante t_i que se encontra no intervalo

$$x_i + \frac{1}{2}\Delta > x(t_i) > x_i - \frac{1}{2}\Delta$$

onde Δ é a diferença entre dois níveis de quantização adjacentes, ou seja o intervalo de quantização. Esta amostra irá ser quantizada pelo nível x_i. Tem-se assim um erro de quantização de

$$\varepsilon_q = x(t_i) - x_i \qquad (2.1)$$

É de salientar que este erro tem uma amplitude limitada a Δ/2 e a amostra $x(t_i)$ tem uma probabilidade idêntica de se situar em qualquer ponto do intervalo referido. Assim, a distribuição do erro de quantização é uniforme, sendo caracterizada pela função densidade de probabilidade

$$p(\varepsilon_q) = \frac{1}{\Delta} \qquad (2.2) \qquad -\frac{\Delta}{2} \leq \varepsilon_q \leq \frac{\Delta}{2}$$

A potência de ruído de quantização pode ser calculada através de

$$Pe = \int_{-\infty}^{\infty} \varepsilon_q^2 p(\varepsilon_q) d\varepsilon_q = \frac{\Delta^2}{12} \qquad (2.3)$$

O efeito cumulativo do ruído de quantização pode ser tratado como ruído aditivo com um efeito similar ao do ruído branco. Deste modo a potência de ruído de quantificação é branco gaussiano e é independente da frequência de amostragem (fs). Assim pode dizer--se que a densidade espectral de potência de ruído é uniforme na frequência, com amplitude K_x como se encontra representado na Figura 2.6.
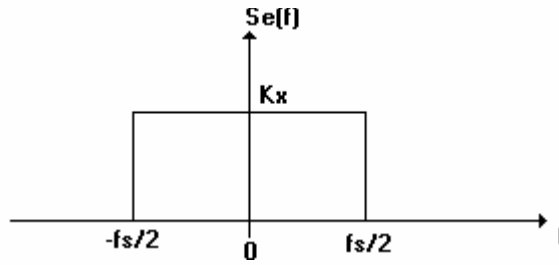
Figura 2.6 – Densidade espectral do ruído de quantização

$$\int_{-fs/2}^{fs/2} Se^2(f) df = \int_{-fs/2}^{fs/2} K_x^2 df = K_x^2 fs = \frac{\Delta^2}{12}$$

$$K_x = \left(\frac{\Delta}{\sqrt{12}}\right)\sqrt{\frac{1}{fs}} \qquad (2.4)$$

Define-se taxa de sobreamostragem (OSR) como

$$OSR = \frac{fs}{2f_0} \qquad (2.5)$$

em que f_0 é a largura de banda do sinal de interesse e $fs > 2f_0$.

Assumindo que o sinal de entrada é um sinal sinusoidal com valor máximo de pico sem distorção de $2^N(\Delta/2)$, em que N representa o numero de bits do quantizador, podemos dizer que a sua potencia é

$$Ps = \left(\frac{\Delta 2^N}{2\sqrt{2}}\right)^2 = \frac{\Delta^2 2^{2N}}{8} \qquad (2.6)$$

Depois da quantização, o sinal $y_1(n)$ é filtrado por H(f) como se encontra representado na figura, obtendo-se o sinal $y_2(n)$. O filtro H(f) elimina o ruído de quantização para frequências superiores a f_0.
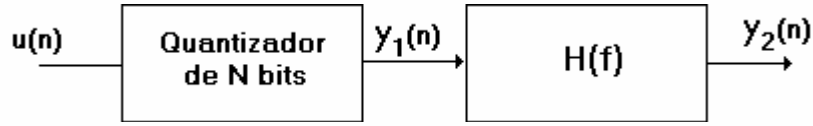
Figura 2.7 – Sistema de eliminação do ruído de quantização para frequências superiores a f_0
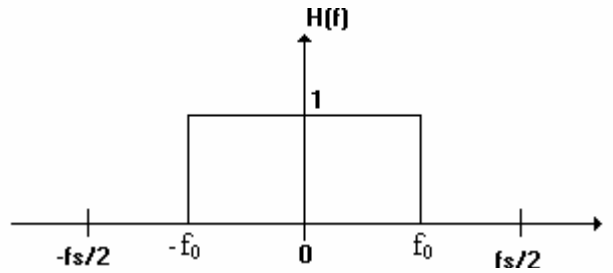
Figura 2.8 – Função de transferência do filtro

Admitindo que o sinal de entrada é de banda limitada, a potência do sinal $y_2(n)$ é igual à potência do sinal de entrada u(n), contudo a potencia do ruído de quantização diminui, diminuição essa provocada pela OSR, como se pode observar pela seguinte expressão.

$$Pe = \int_{-fs/2}^{fs/2} Se^2(f)\left|H(f)\right|^2 df = \int_{-f_0}^{f_0} K_x^2 df = \frac{2f_0}{fs}\frac{\Delta^2}{12} = \frac{\Delta^2}{12}\left(\frac{1}{OSR}\right) \qquad (2.7)$$

Assim, de cada vez que se duplica a OSR, a potência do ruído de quantização diminui para metade.

Neste momento já é possível calcular a SNR, visto já se possuir a potência do sinal de entrada (Ps) e a potência do ruído de quantização (Pe). Assim vem

$$SNR = 10\log\left(\frac{Ps}{Pe}\right) = 10\log\left(\frac{3}{2}2^{2N}\right) + 10\log(OSR)$$

$$SNR = 6.02N + 1.76 + 10\log(OSR) \qquad (2.8)$$

Agora pode-se concluir que a sobreamostragem melhora a SNR de 3dB/oitava, o que equivale a 0.5bits/oitava. Apesar do nível médio de ruído ter diminuído não implica que a energia de ruído total não seja a mesma, apenas agora está distribuída por uma largura de banda superior.

Os conversores Delta-Sigma aproveitam o facto de a energia total de ruído estar mais espalhada para, através de um filtro digital, eliminar uma grande quantidade de ruído.

Por exemplo, quantos bits são garantidos usando uma taxa de sobreamostragem de 4?

Por cada taxa de sobreamostragem de 4 vezes, está-se a incrementar a SNR de 6dB, então com um bit ADC apenas obtemos dois bits. Então qual é a taxa de sobreamostragem para garantir 24 bits? Temos que ter uma taxa de 4^{23}, o que é claramente impraticável.

Os ADC contornam esta limitação conseguindo aumentar a SNR acima dos 6dB por cada taxa de sobreamostragem de 4 vezes. Para se conseguir isso utiliza-se o seguinte diagrama de blocos.
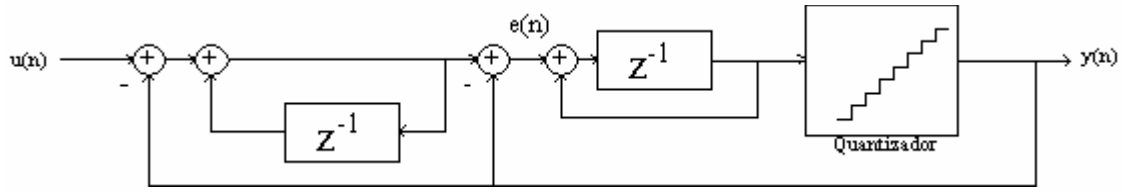
Figura 2.9 – Diagrama de Blocos de um Modulador Delta-Sigma de 2ª ordem

O diagrama de blocos acima é um modulador de 2ª ordem. A função de transferência do ruído, $N_{TF}(z)$ é uma função passa-alto de 2ª ordem.

A função de transferência do sinal é

$$S_{TF}(f) = \frac{Y(f)}{U(f)} = z^{-1} \qquad (2.9)$$

e a de ruído é dada por

$$N_{TF}(f) = \frac{Y(f)}{E(f)} = \left(1 - z^{-1}\right)^2 \qquad (2.10)$$

Fazendo $z=e^{jwT}=e^{j2\pi f/fs}$, obtem-se para o modulo da função de transferência do ruído, um filtro passa alto, com

$$|N_{TF}(f)| = \left(2\sin\left(\frac{\pi f}{fs}\right)\right)^2 \qquad (2.11)$$

resultando numa potencia do ruído de quantização para frequências superiores à banda de interesse de

$$Pe = \int_{-f_0}^{f_0} Se^2(f)\left|N_{TF}(f)\right|^2 df = \int_{-f_0}^{f^0} K_x^2\left|N_{TF}(f)\right|^2 df = \int_{-f_0}^{f^0}\left(\frac{\Delta^2}{12}\right)\frac{1}{fs}\left[2\sin\left(\frac{\pi f}{fs}\right)\right]^4 df$$

$$Pe \cong \frac{\Delta^2\pi^4}{60}\left(\frac{1}{OSR}\right)^5 \qquad (2.12)$$

No modulador de 2ª ordem consegue-se uma SNR de

$$SNR = 10\log\left(\frac{Ps}{Pe}\right) = 10\log\left(\frac{3}{2}2^{2N}\right) + 10\log\left[\frac{5}{\pi^4}(OSR)^5\right]$$

o que equivale a

$$SNR = 6.02N + 1.76 - 12.9 + 50\log(OSR) \qquad (2.13)$$

Pode-se concluir que quando se duplica a OSR melhora-se a SNR para o modulador de 2ª ordem de 15dB, ou seja de 2.5bits/oitava.

A chave para o sucesso do modulador é um integrador. O integrador opera como um filtro passa-baixo para o sinal de entrada e como um filtro passa-alto para o ruído de quantização. Em consequência disso grande parte do ruído de quantização aparece nas altas frequências.

Usando este tipo de moduladores é possível implementar um ADC delta-sigma com uma taxa de sobreamostragem razoável.

O objectivo principal do filtro digital e do filtro decimador é extrair os bits de saída do fluxo de dados e reduzir o débito binário para valores mais usuais.

Como é que o filtro decimador consegue reduzir o débito binário proveniente do filtro digital?

Ele apenas aproveita algumas amostras de entrada. Por exemplo se existe uma taxa de decimação de 4, então só a cada 4 amostras de entrada é que ele aproveita uma e descarta as restantes. Na prática, ambas as taxas dos filtros estão relacionadas.

2.4.1 Conclusão

Existe um conjunto de parâmetros que se encontram interrelacionados entre si. Isto é, é completamente impossível escolher um desses parâmetros sem afectar os restantes. As relações de compromisso que devem existir entre os parâmetros são os seguintes:

- Largura de banda, resolução, ganho, taxa de sobreamostragem, ritmo de amostragem, sinal de entrada.

Para se entender melhor a relação que existe entre todos os parâmetros, dá-se o seguinte exemplo:

Quando o sinal de entrada aumenta o ganho diminui, ou se o ganho aumenta a resolução diminui.

2.5 Arquitectura de Controlo

O objectivo do trabalho não incluía nenhum projecto de arquitectura de tempo real para controlo, pelo que se recorreu à utilizada em diversas aplicações pelo o ISR. Esta arquitectura é implementada pelo microcontrolador PXAS3 e pelo módulo de interface CAN2.0 projectada para controlo em tempo real, onde o tamanho reduzido e baixo consumo de energia são factores de extrema importância.

O microcontrolador é um *Philips* PXAS3 de 16 bits e de elevado desempenho, baseado na arquitectura do conhecido 8051, com as capacidades aumentadas pela introdução de vários periféricos destinados a satisfazer as necessidades provenientes de

novas aplicações. A velocidade desta arquitectura, XA, é cerca de 10 a 100 vezes a do 8051.

Características Principais:

- Compatível com 8051;
- 24 bits endereçáveis;
- Rápidas instruções de multiplicação e divisão;
- Elevado desempenho de conversão (Conversor Analógico-Digital a 8 bit com 8 canais);
- Frequência de oscilação acima de 30MHz para 2.7V-5.5V de tensão de alimentação;
- Ciclo de Instrução de 100ns;
- Interface I^2C;
- 50 pinos de Entrada/Saída com 4 configurações de saída programáveis;
- *Watchdog timer*;
- Temperatura de funcionamento de 0 a 70˚C;
- 7 vectores de interrupção por *software*.

Conector

A PCI utilizada tem uma grande quantidade de conectores os quais permitem a ligação a diversos periféricos. De seguida será só apresentado o conector com interesse para a execução do projecto, Conector Analógico – P1.

Conector Analógico – P1

O subsistema analógico é capaz de adquirir oito diferentes sinais analógicos. Esses sinais e alguns dos sinais adicionais podem ser encontrados no Conector P1. A função de cada pino neste Conector é detalhado na tabela que se segue:

Pino	Função
1	
2	Referência Baixa Opcional
3	
4	
5	Massa Analógica
6	AD0
7	AD2
8	AD4
9	AD6
10	Massa Analógica
11	Tensão analógica de alimentação opcional
12	
13	
14	
15	Massa Analógica
16	AD1
17	AD3
18	AD5
19	AD7
20	Referência Alta

Tabela 2.1 – Funções dos pinos do Porto P1

2.6 Conversor Analógico-Digital – ADS1210

Face ao objectivo estabelecido no Trabalho Final de Curso, o componente Conversor Analógico-Digital ADS1210 desempenha uma função preponderante na sua realização. Devido a esse facto será alvo de um estudo mais profundo no decorrer do presente capítulo.

2.6.1 Breve Descrição

Sucintamente o Conversor Analógico-Digital ADS1210 é um conversor de sinais analógicos para sinais digitais, projectado para aplicações em que é requerida uma elevada resolução e construído de acordo com a arquitectura Delta-Sigma.

Cada ADS1210 é constituído por um amplificador de ganho programável (PGA), um modulador Delta-Sigma de segunda ordem, um filtro digital programável, um microcontrolador, um circuito gerador de relógio e uma tensão de referência.
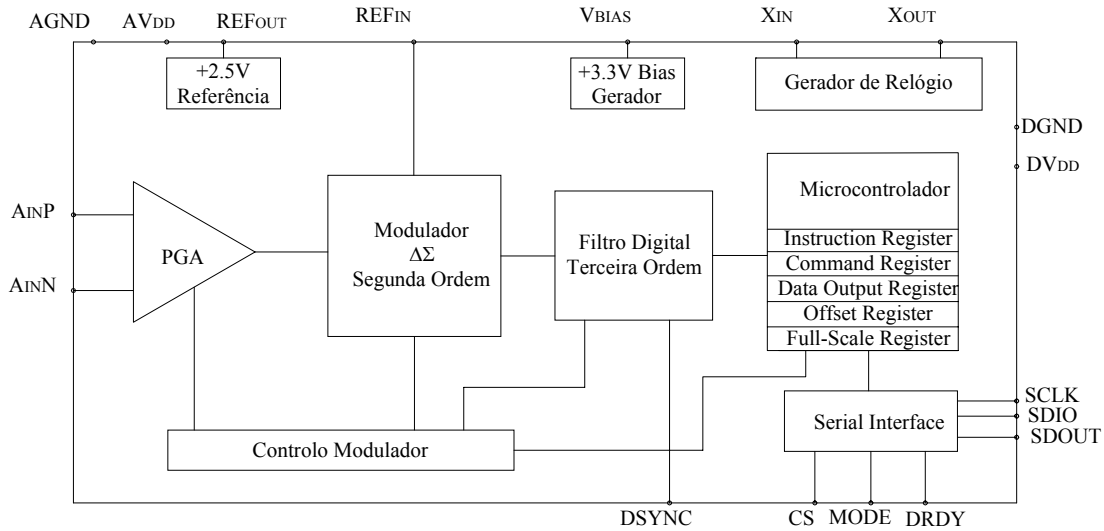
Figura 2.10 – Diagrama de Blocos de um ADS1210

O *Turbo Mode* (TM) é uma das ferramentas do ADS1210 que pode ser usada para aumentar a frequência de amostragem na entrada do condensador do PGA, a qual é normalmente 19.5KHz com um relógio de 10MHz. Através da programação do *Command Register*, registo incluído no microcontrolador do conversor, a frequência de amostragem pode ser aumentada para 39KHz, 78KHz, 156KHz ou 312KHz, correspondendo a cada aumento um crescimento no desempenho do ADS1210, quando mantida a mesma taxa de dados. Por sua vez, o PGA pode tomar valores para o seu ganho de 1,2,4,8 ou 16. Este ganho é implementado pelo aumento do número de amostras tomadas à entrada do condensador, a partir de 19.5KHz para o caso de ganho 1 e a 312KHZ para o caso de ganho 16. Então, pode dizer-se que as funções TM e PGA são ambas implementadas pelo aumento da frequência de amostragem à entrada do condensador encontrando-se a combinação limitada a 16. Seguidamente apresenta-se a expressão que permite determinar a frequência de amostragem na entrada do condensador:

$$f_{SAMP} = \frac{f_{XIN} \times TurboMode \times Gain}{512} \qquad (2.14)$$

Quanto à taxa de dados de saída ou simplesmente taxa de dados, esta pode variar de poucos hertz até 15.625KHz, resultando que o aumento da taxa de dados é acompanhado pela diminuição do desempenho do ADS1210. Mudanças na taxa de dados não afectam a frequência de amostragem à entrada do condensador.

O ADS1210 também inclui um quadro completo de calibrações que são utilizadas para correcções de erros de ganho ou para limitar os erros internos do sistema. As várias calibrações possíveis, *Offset Calibration*, *Full-Scale Calibration*, *Self-Calibration, Pseudo System Calibration e Background Calibration,* podem ser feitas sempre que for necessário ou de um modo automático e continuo. Os registos de calibração podem ser lidos e escritos, permitindo assim a comutação entre diferentes configurações – diferentes escolhas de TM, de ganho, taxa de dados, etc. Contudo, em

todos os casos uma nova calibração é sempre necessária. O melhor desempenho do ADS1210 é conseguido com a execução da *Offset Calibration*, *Full-Scale Calibration* e *Self-Calibration* seguindo esta mesma ordem. Cada um dos vários tipos de calibração é programado através do registo *Command Register*.

Os vários parâmetros, modos, configurações e registos do ADS1210 são lidos e escritos via *synchronous serial interface*. Esta interface pode operar em dois modos: o modo de relógio próprio, ou *Master Mode*, em que a *serial clock frequency* (SCLK) é limitada a metade da frequência de relógio (*System Clock Input* - X_{IN}) do ADS1210; e o modo de relógio externo, ou *Slave Mode*, em que SCLK se encontra limitada a X_{IN} a dividir por cinco. Esta é uma importante consideração para muitos sistemas e pode determinar o valor máximo de X_{IN} do ADS1210 que pode ser usada. A utilização do ADS1210 em *Master Mode* implica a execução de apenas uma instrução por ciclo de conversão. O modo *Master Mode* é activo quando o pino de entrada do ADS1210 denominado *Mode* se encontra no estado lógico um. Pelo contrário, o modo *Slave Mode* é activo quando o mesmo pino de entrada se encontra no estado lógico zero e permite a execução de mais do que uma instrução por ciclo de conversão. Por outro lado em *Slave Mode* o pino de entrada denominado *Chip Select Input* (CS) é utilizado como comutador, isto é, permite ou inibe a comunicação com o ADS1210, ao contrário do sucedido no caso de operar em *Master Mode* em que o pino de entrada CS é utilizado para não permitir a leitura mesmo com o pino de saída *Data Ready* (DRDY) no estado lógico baixo até o *main controller* poder fornecer a comunicação. O modo *Master Mode* é também caracterizado pela utilização do pino SCLK configurado como pino de saída, o que poderá colocar problemas a muitos microcontroladores que controlam a comunicação com o ADS1210, particularmente quando a X_{IN} é maior do que poucos MHz, pois SCLK poderá exceder o máximo da SCLK do microcontrolador.

O microcontrolador do ADS1210 consiste numa *Arithmetic Logic Unit* (ALU) e um banco de registos. O microcontrolador tem dois estados: *power-on reset* e de conversão. No primeiro destes estados, o microcontrolador executa o *power-on reset* a todos os seus registos para o seu estado de defeito e executa a *Self-Calibration* para uma taxa de dados de 850Hz. Depois disto, o microcontrolador entra no estado de conversão o qual é considerado como o estado normal de operação do ADS1210. Os registos *Instruction Register* (INSR) e *Command Register* (CMR), com 8 bits e 32 bits respectivamente, controlam a operação de conversão. O registo *Data Output Register* (DOR), de 24 bits, contém o resultado da mais recente conversão. Por último os registos *Offset Calibration Register* (OCR) e *Full-Scale Calibration Register* (FCR), de 24 bits cada um, contêm dados usados para correcção do resultado da conversão interna, antes que o resultado da conversão seja enviado para o registo DOR. Os dados destes dois últimos registos devem ser o resultado de uma rotina de calibração ou devem ser valores que tenham sido escritos directamente através de *synchronous serial interface*. È através da instrução que é escrita no INSR que se determina qual o tipo de comunicação que ocorrerá (leitura ou escrita). Por sua vez, a escrita do CMR permite controlar toda a funcionalidade do ADS1210. Este registo inclui a escolha do ganho, TM, taxa de dados, formato dos dados, modo de operação, etc. Por fim, o DOR, o qual contém o resultado da mais recente conversão é alterado com um novo resultado, mas só depois do pino de saída denominado DRDY ir para o estado lógico baixo. Se o resultado contido pelo DOR não for lido dentro de um período de tempo definido por $\frac{1}{f_{DATA}} - 12 \times \left(\frac{1}{f_{XIN}} \right)$, então o novo resultado de conversão sobrepor-se-á ao antigo, ou seja, substituirá o antigo resultado.

Os pinos denominados de *Serial Data Input* (SDIO) e *Serial Data Output* (SDOUT) do ADS1210 constituem respectivamente o modo de acesso entre o utilizador e a *synchronous serial interface* para poder executar a escrita e leitura dos vários registos que fazem parte do microcontrolador do ADS1210. Contudo, este permite que apenas um pino seja utilizado para a execução de ambas as funções, o pino SDIO, desde que devidamente configurado através do CMR.

O filtro digital do ADS1210 é de terceira ordem, que conduzirá a resultados de saída baseados no mais recente resultado obtido a partir do modulador Delta-Sigma. O número de resultados que são usados depende do conjunto *decimation ratio* (taxa de decimação) inserido no CMR. De uma forma simplista o filtro digital pode ser interpretado como uma simples média de resultados do modulador e apresentação dessa média como dados digitais de saída.
O filtro digital é descrito pela seguinte função de transferência

$$|H(f)| = \left| \frac{\sin\left(\frac{\pi \times f \times N}{f_{MOD}}\right)}{N \times \sin\left(\frac{\pi \times f}{f_{MOD}}\right)} \right|^3, \quad (2.15)$$

onde N é o *Decimation Ratio* e f_{MOD} é a frequência à qual o modulador está a trabalhar.
O filtro tem como resposta $\left(\frac{\sin(x)}{x}\right)^3$ a que corresponde um filtro $sinc^3$.

A taxa de dados é um termo dependente das configurações pretendidas, mais directamente da X_{IN} e da escolha do TM. Quanto maior for a escolha para o TM mais rápido será o funcionamento do modulador. A taxa de dados é dada pela seguinte expressão:

$$f_{DATA} = \frac{f_{XIN} \times TurboMode}{512 \times (DecimationRatio + 1)} \quad (2.16),$$

na qual o factor *Decimation Ratio* determina o número de resultados do modulador que são usados pelo filtro digital para calcular cada resultado de conversão. Este factor encontra-se limitado pelo intervalo de 19 a 8000. Fora deste intervalo o filtro digital conduzirá a resultados incorrectos.

O número de resultados do modulador, usados para se calcular cada resultado de conversão é três vezes o *Decimation Ratio*. Isto significa que, qualquer alteração introduzida no ADS1210 requererá de pelo menos três ciclos de conversão até se conseguirem resultados válidos. Cada ciclo de conversão é entendido como uma transição completa do pino de saída DRDY, isto é, do nível lógico zero para o nível lógico um e novamente para o nível lógico zero, ou vice-versa. No entanto, e conforme já referido, aquando de alguma alteração, só pelo menos à quarta passagem do pino de saída DRDY pelo nível lógico zero se considera existirem resultados de conversão válidos.

Outro pino de grande importância na constituição do ADS1210 é o pino de entrada denominado de *Control Input to Synchronize Serial Output Data* (DSYNC). Este pino permite a sincronização de múltiplos conversores. Esta funcionalidade é de grande interesse quando se utilizam vários conversores e/ou se pretende que todos forneçam resultados de conversão referidos ao mesmo instante. Isto consegue-se através

da transição do pino DSYNC de o nível lógico um para o nível lógico zero, onde terá de permanecer no mínimo 10,5*t_{XIN} e novamente para o nível lógico um. Outra funcionalidade é executar o *reset* ao modulador para zero de modo a obter-se dados válidos o mais rapidamente possível aquando de alguma mudança. Estas duas funcionalidades também podem ser desempenhadas por um bit denominado DSYNC existente no CMR, desde que devidamente configurado.

No conversor ADS1210 as suas duas entrada para sinais analógicos podem ser configuradas para utilização em dois modos distintos: modo comum e modo simétrico. A existência destes dois modos permite a aplicação de sinais unipolares de 0 a 5volt e de sinais bipolares de ±10volt. O modo simétrico caracteriza-se pela utilização da saída V_{BIAS} e pelo uso de dois pares de resistências com um factor comum de 3 na relação entre eles.
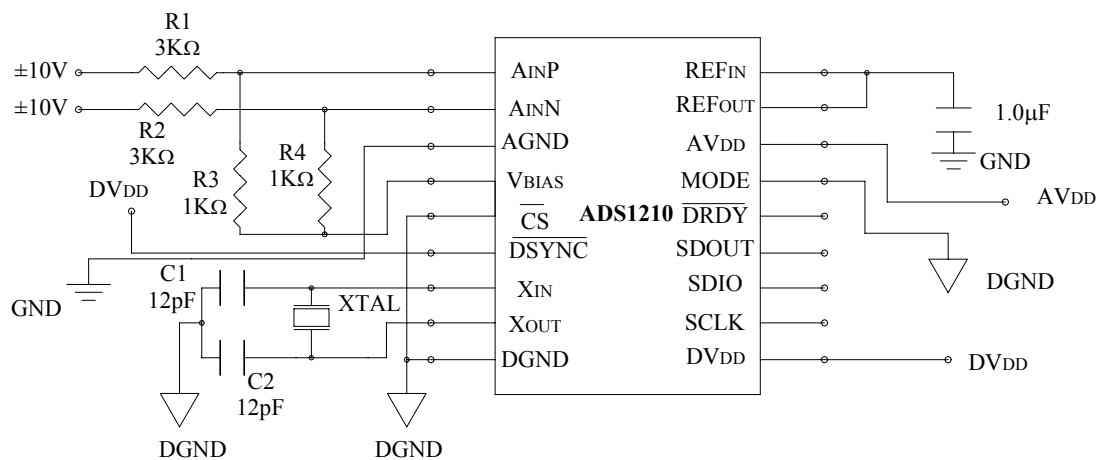
Figura 2.11 – Esquema de ligações de entradas $A_{IN}P$ e $A_{IN}N$ para modo simétrico

O aumento dos valores destes pares de resistências conduz a uma redução da potência de dissipação. Contudo, em todos os casos, a corrente máxima dentro ou fora de V_{BIAS} não deve exceder a especificação de 10mA.

Além dos modos de operação já mencionadas (calibração e conversão), o ADS1210 também se pode encontrar no modo *Sleep Mode*. Tal como nos outros casos, este modo é também activado através de uma configuração específica do CMR, bit MD2-MD0, e excitado através de uma nova configuração introduzida na mesma localização da anterior. O modo *Sleep Mode* permite a redução da potência dissipada por parte do ADS1210. Sempre que este modo é activado tanto a saída *Bias Voltage Output* (V_{BIAS}) como a referência interna devem ser desactivadas através da configuração dos bits BIAS e REFO do CMR. Para se iniciar a comunicação com o conversor quando este se encontra em *Sleep Mode* um dos procedimentos seguintes deve ser efectuado: caso o pino de entrada CS esteja a ser controlado, deve-se levar esta entrada ao estado lógico zero após o qual a comunicação será restabelecida normalmente; caso o pino CS esteja a actuar de modo amarrado (fixo) e o ADS1210 se encontre a operar em *Master Mode* então a transição de estado lógico deve ser feita ao nível da linha SDIO. Se SDIO se encontrar no estado lógico zero, a linha SDIO deve ser levada ao estado lógico um e aí permanecer no mínimo durante 2*t_{XIN} regressando após

isso novamente ao estado lógico zero. Alternativamente, SDIO pode ser forçado a ir para o estado lógico um depois do ADS1210 entrar em *Sleep Mode*, e quando ele está para ser excitado, a linha SDIO é levada ao estado lógico zero. Por último se CS está fixo e o ADS1210 está em *Slave Mode*, então o simples envio de um comando para o INSR restabelecerá a comunicação. Quando um novo modo é activado, exceptuando o modo *Sleep Mode*, o ADS1210 executará uma sequência interna para activar o circuito analógico e o circuito digital. Uma vez esta sequência executada, um ciclo normal de conversão é realizado antes que o novo modo tome efeito. Após isto o conversor responderá normalmente. O sinal DRDY permanecerá no estado lógico um durante o primeiro ciclo de conversão e assim permanecerá no segundo, a menos que o novo modo de operação seja o seu modo normal, ou seja, o modo de conversão.

Um ponto a ter em atenção aquando da alimentação do conversor ADS1210 é que este requer que a alimentação do circuito digital não seja superior à do circuito analógico +0,3volt. Na maioria dos sistemas, isto significa que o circuito analógico arranque primeiro que o circuito digital. Outra preocupação a ter-se em conta é que entradas do ADS1210 como SDIO, *Noninverting Input* ($A_{IN}P$), *Inverting Input* ($A_{IN}N$) ou *Reference Input* (REF_{IN}) não devem estar presentes antes de os circuitos analógico e digital estejam correctamente alimentados.

2.7 Cálculo de Resolução Efectiva

Resolução efectiva ou número efectivo de bits (ENOB) é simplesmente a razão do número de pontos de dados com frequência correcta (original) e o número de pontos de dados com outros valores de frequência. Esta razão é expressa em bits.

ENOB diminui com o aumento da frequência de taxa de conversão requerida. A Figura 2.12, e de acordo com Dr. Jerry Horn[1], mostra a dificuldade em estimar a representação digital de um dado sinal analógico versus a frequência de taxa de conversão.
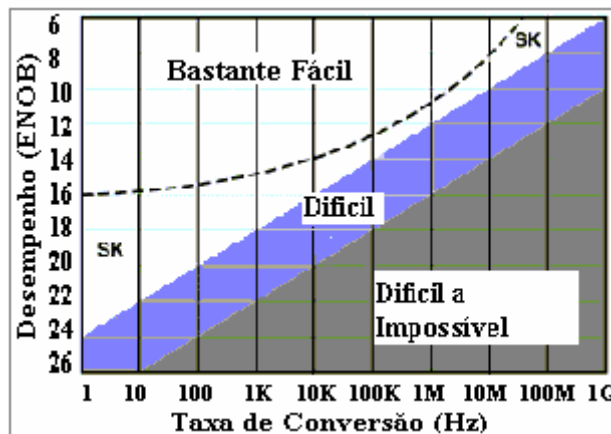
Figura 2.12 – Resolução Efectiva versus da frequência de taxa de conversão

Existem vários métodos para o cálculo de ENOB, o qual é função da relação sinal ruído (SNR) do conversor Analógico-Digital (ADC).

$$SNR = ((ENOB \times 6.02) + 1.76)dB \qquad (2.17)$$

Os mais comuns são os métodos recorrendo à utilização de uma onda sinusoidal e os testes com transformada rápida de Fourier (FFT), surgindo também, cada vez mais comum e como uma alternativa efectiva, o Método do Histograma.

O método do histograma consiste na conversão de um sinal periódico, bem conhecido com o ADC sob teste, em tempos de amostragem que são assíncronos relativamente ao sinal de entrada. Os dados são vistos na forma de um histograma normalizado mostrando a frequência de ocorrência de cada um. O problema aparece quando a forma da onda aplicada apresenta distorção relativamente ao caso ideal. De facto, é muito difícil gerar uma onda triangular com uma linearidade perfeita ou uma onda sinusoidal com muito poucas harmónicas. Contudo, ENOB pode ser calculado analisando o histograma dos dados digitalizados e portanto, apenas o conhecimento da função distribuição do sinal de entrada, melhor do que a sua estrutura no domínio do tempo, é requerido.

A sua estrutura no domínio do tempo pode não ser de todo conhecido, mas é necessário que a sua função distribuição de probabilidade o seja. O histograma ideal será então comparado com um gerado com os valores digitalizados e poder-se-á calcular a ENOB do conversor.

Este método requer a aplicação de um sinal de entrada, com uma função distribuição bem conhecida ao ADC em teste. Aqui abordaremos dois casos para o sinal de entrada: fonte de ruído e fonte sinusoidal.

Para o caso de uma fonte de ruído a função densidade de probabilidade contínua é dada pela expressão (2.18) e onde deve ser convertida numa função densidade de probabilidade discreta, a qual fornece a probabilidade de uma amostra digitalizada pela palavra de código binária *i* para um ADC bipolar *n*-bit.

$$p(V) = \frac{1}{\sqrt{2\pi} \times A} \exp\left(- \frac{V^2}{2 \times A^2} \right) \qquad (2.18)$$

onde A é a amplitude de desvio padrão do sinal e V representa a variável aleatória que é a tensão de entrada do sinal. Amostragem assíncrona do sinal de entrada não é requerida, dado a sua natureza aleatória e não periódica.

O histograma ideal por sua vez é dado pela expressão seguinte:

$$P(i) = \frac{1}{\sqrt{2\pi} \times \sigma} \left(\exp\left(- \frac{\left(\frac{i - 2^{n-1}}{2} \right)^2}{2 \times \sigma^2} \right) \right) \qquad (2.19)$$

com $\sigma = \left(\frac{A}{Vref} \right) \times 2^n$, e onde Vref é o intervalo dinâmico em toda a extensão do ADC, também designado de *full-scale dynamic range* (V_{FSR}).

A tensão limiar para o nível *i* é calculada de acordo com a expressão seguidamente apresentada:

$$V(i) = \frac{\sqrt{-2 \times \sigma^2 \ln\left(P(i) \times \sigma \times \sqrt{2\pi} \right)}}{2^{n-1}} \times Vref \qquad (2.20)$$

onde o sinal V(i) para $i<2^{n-1}$ deve ser mudado. Esta é uma expressão muito usual porque ela basicamente fornece a característica de quantização de um ADC.

Para o caso de uma fonte sinusoidal, caso mais comum, utiliza-se uma onda sinusoidal bem conhecida com amplitude A,

$$V = A \times \sin(\omega t) \qquad (2.21)$$

e cuja função densidade de probabilidade é

$$p(V) = \frac{1}{\pi \times \sqrt{(A^2 - V^2)}} \qquad (2.22)$$

Integrando esta expressão em ordem à tensão, obtêm-se a função distribuição:

$$P(Va, Vb) = \frac{1}{\pi}\left(a\sin\left(\frac{Vb}{A}\right) - a\sin\left(\frac{Va}{A}\right)\right) \qquad (2.23)$$

a qual considera probabilidade da amostra estar no intervalo [Va , Vb].

Para um ADC com intervalo dinâmico Vref, e Va-Vb=1, convertendo a função distribuição de probabilidade contínua em discreta obtêm-se:

$$P(i) = \frac{1}{\pi}\left(a\sin\left(\left(\frac{2i - 2^n - 1}{2^n}\right) \times \frac{Vref}{A}\right) - a\sin\left(\left(\frac{2i - 2^n - 3}{2^n}\right) \times \frac{Vref}{A}\right)\right) \qquad (2.24)$$

Isto é, a probabilidade de uma amostra representada por meio de uma palavra de código binária i para uma onda sinusoidal de entrada de amplitude A.

A tensão limiar para o nível *i* é calculada pela expressão seguinte:

$$V(i) = -A \times \cos\left(\frac{\pi \times CH(i)}{N}\right) \qquad (2.25)$$

onde N é o número total de amostras tomadas, e CH(i) é o histograma acumulativo definido por:

$$CH(i) = \sum_{j=0}^{i-1} H(j) \text{ e } CH(0)=0$$

Dada a tensão limiar para o nível *i*, V(i), a potência média de quantização é

$$\sigma_q = \frac{1}{3(2^n \times \Delta)} \times \sum_{i=1}^{2^n} li\left(a_i^2 + a_i b_i + b_i^2\right) \qquad (2.26)$$

sendo Δ o passo de quantização, e

$$a_i = \left(-2^{n-1} + 1\right) \times \Delta - V(i) \qquad (2.27)$$

$$b_i = \left(-2^{n-1}+1\right)\times\Delta - V(i+1) \qquad (2.28)$$

$$l_i = V(i+1) - V(i) \qquad (2.29)$$

O número efectivo de bits, n1, é então calculado por:

$$n1 = n - \log_2\left(\frac{\sqrt{12}\times\sigma_q}{\Delta}\right) \qquad (2.30)$$

As expressões (2.19) e (2.25) são usadas em (2.27), (2.28) e (2.29) para calcular ENOB (n1), quando fonte de ruído e fonte sinusoidal são usadas.

Isto é, de uma forma simples e experimental, calcular ENOB de um ADC é medir SNR. Para isso começa-se por injectar um sinal sinusoidal com o menor número de harmónicas possíveis e à recolha de *m* dados (em que *m* deve ser um número suficientemente grande, da ordem dos milhares) a partir da placa de aquisição de dados. Seguidamente aplica-se FFT a esse conjunto de dados. A FFT converte pontos de dados no domínio do tempo para o domínio da frequência, efectivamente operando como um analisador espectral. Os *m* dados originais recolhidos são assim convertidos em *m* valores separados de frequência. Então através da sua análise é possível medir SNR. A SNR é a razão do valor eficaz do sinal de entrada presente na saída e o valor eficaz de todas as outras componentes espectrais presentes na saída que se encontram abaixo da frequência de nyquist, excluindo componentes DC.

Neste momento e de acordo com a expressão (2.17) é possível calcular ENOB.

Para o caso especifico do conversor em estudo, ADS1210, o seu *datasheet* fornece duas maneiras simples de calcular a sua resolução efectiva, uma dando o resultado em bits rms (referenciados à saída) e a outra em microvolt rms (referenciado à entrada). No entanto ambas têm a desvantagem de dependerem uma da outra, levando por isso à necessidade de se ter conhecimento de pelo menos uma delas.

$$ERinbitsrms = \frac{20\times\log\left(\frac{\left(\frac{10}{PGA}\right)}{ERinvrms}\right) - 1.76}{6.02} \qquad (2.31)$$

$$ERinvrms = \frac{\left(\frac{10}{PGA}\right)}{10^{\left(\frac{6.02\times ERinbitsrms + 1.76}{20}\right)}} \qquad (2.32)$$

Isto conduz a que quando não se sabe nenhuma destas grandezas tem-se de recorrer a um dos métodos referidos em cima.

3 Descrição da Arquitectura de Aquisição de Sinais

3.1 Introdução

Neste capítulo pretende-se fazer uma explicação detalhada de como os diversos blocos constituintes da placa de aquisição de sinais (PAS) foram interligadas. Na Figura 1 encontra-se um diagrama exemplificativo de como se interliga todo o sistema de aquisição de sinais, e de como os sinais eléctricos flúem pelo sistema. Na Figura 2 encontra-se um diagrama que representa os principais blocos desta placa e as ligações de cada um deles. O seu objectivo é fornecer uma visão geral do que irá ser explicado de seguida.

A leitura deste capítulo deve ser acompanhada do desenho do esquemático, que se encontra no anexo A.
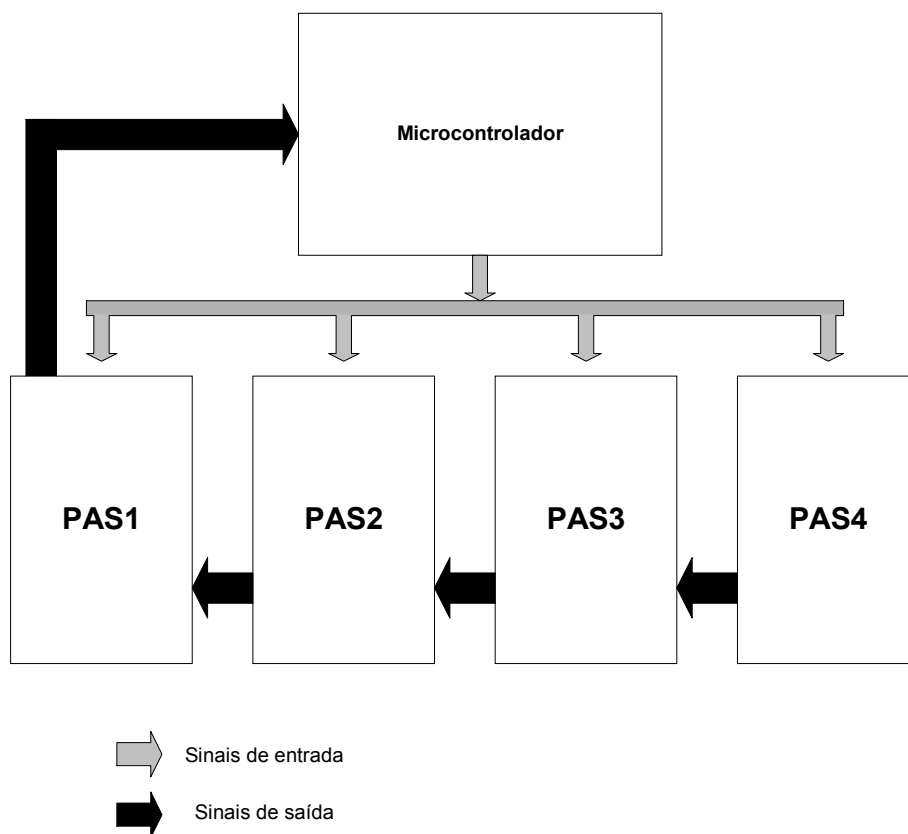
Figura 3.1 – Interligação do Sistema

Cada PAS encontra-se dividida em duas partes completamente distintas, a parte analógica e a digital. Esta separação resulta da necessidade de reduzir ao máximo o ruído existente no circuito e de o ADC necessitar de duas fontes de tensão distintas, em que a parte analógica deve ser primeiro ligada e só depois a digital.

A separação entre a parte analógica e a parte digital é feita ao nível dos conversores, pela sua própria separação.
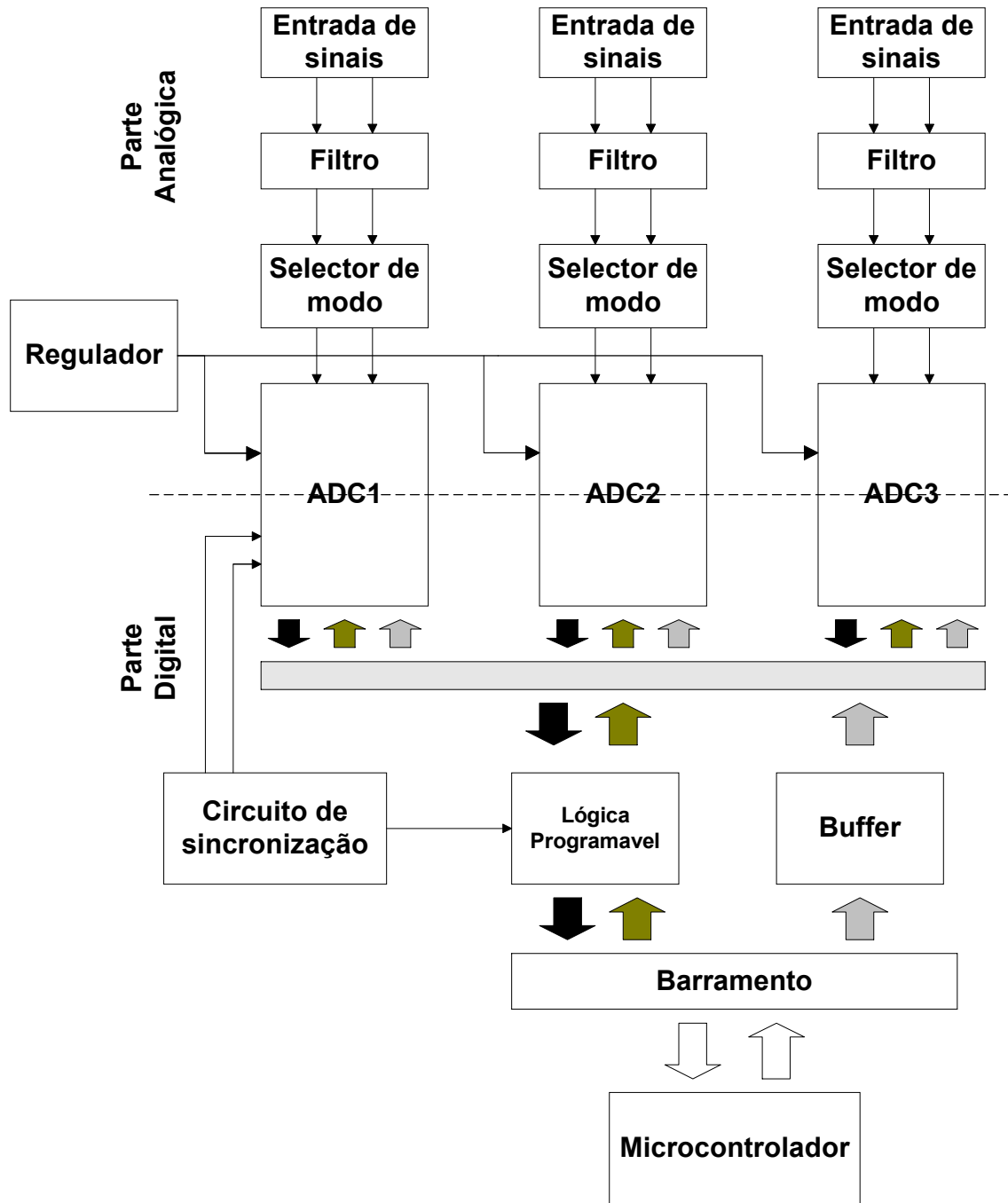
Figura 3.2 – Diagrama de Blocos de uma PAS

3.2 Implementação da Parte Analógica

3.2.1 Regulador

A parte analógica da PAS funciona com uma tensão de alimentação de +5V. Como já foi referido esta fonte tem que ser de grande qualidade, isto é, bem regulada e estabilizada e de muito baixo ruído.

O ruído é um factor muito importante neste tipo de circuitos, porque pode influenciar grandemente a SNR e consequentemente a resolução efectiva do conversor.

Neste trabalho utilizou-se um regulador, versão fixa a +5V da Burr-Brown, que tem uma corrente máxima de saída de 1A e é de muito baixo ruído. Á entrada de AVDD do ADC ainda existe um condensador de desacoplamento.
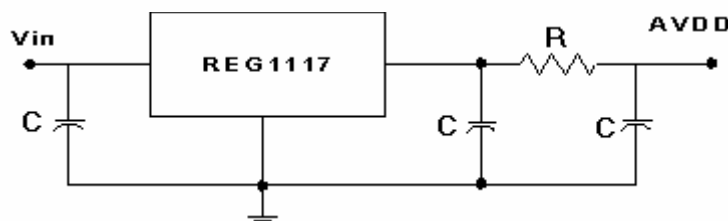
Figura 3.3 – Esquema de Blocos de ligação do Regulador

3.2.2 Filtro

De acordo com o teorema de Nyquist um sinal deve ser amostrado a uma frequência igual ou superior ao dobro da frequência máxima do sinal, para que este possa ser reproduzido integralmente sem erro de "aliasing", ou seja, sem deformação do sinal original.

Dado as características do ruído adicionado pelos componentes electrónicos, vibrações, etc, não é possível garantir que o sinal de entrada não contenha sinais acima de metade da frequência de amostragem, o que torna necessário filtrar o sinal com um filtro passa baixo com frequência de corte igual ou menor a frequência de Nyquist.

Tendo em vista a eliminação do "aliasing" projectou-se um filtro passa-baixo, de modo a que o sinal seja fortemente atenuado para as altas frequências e deste modo aumentar a resolução do sistema.

Este filtro é utilizado nas duas entradas de sinal, uma chamada de entrada não inversora e a outra de entrada inversora. Ainda se utilizou um condensador de desacoplamento entre estas duas entradas de sinal.

É de referir que os componentes utilizados nesta parte de circuito são de grande precisão e estabilidade de modo a proporcionar um baixo ruído.
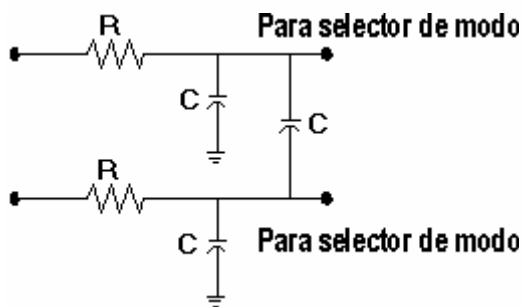
Figura 3.4 – Esquema de ligações de um Filtro

Para os filtros passa-baixo a frequência de corte é dada por:

$$f_c = \frac{1}{2 \times \pi \times R \times C} \qquad (3.1)$$

O filtro está projectado para uma frequência de corte de 15KHz.

A frequência de amostragem depende dos valores atribuídos aos parâmetros do ganho e turbo mode do ADC e é calculada pela expressão (2.14).

Para um ganho de 16, a frequência de amostragem é de 312KHz, o que metade desse valor é muito superior à frequência de corte. Assim evita-se que parte do espectro acima de $f_{mod}/2$ seja dobrada em torno de $f_{mod}/2$ e invertida espectralmente, ou seja, frequências mais altas possam passar por frequências menores e deste modo o sinal reproduzido seja uma cópia deformada do sinal original.

A frequência do filtro interno é calculada pela expressão (2.16), e é no máximo de 1KHz.

Escolheu-se uma frequência de corte uma década acima da frequência do filtro interno devido ao filtro externo passa baixo introduzir atraso de fase.

3.2.3 Selector de Modo

Este selector permite escolher entre o modo unipolar ou o modo bipolar.

Quando o selector está em modo unipolar a saída do filtro passa-baixo está ligada directamente ás entrada não inversora e inversora ($A_{IN}N$ e $A_{IN}P$, respectivamente) do ADC. Este modo é caracterizado pela permissão de sinais de entrada com um máximo de 5V e um mínimo de 0V. Esta limitação implica se ter uma *Voltage Full-Scale Range* (V_{FSR}) no máximo de 10V, como se pode observar pela Tabela 3.1.

Se o selector estiver no modo bipolar as saídas dos filtros encontram-se ligadas às entradas $A_{IN}N$ e $A_{IN}P$ do ADC através de um circuito resistivo, como o apresentado na Figura 3.5. A tensão de saída do ADC, V_{BIAS} é dependente da tensão de referência interna (REF_{IN}) do ADC e é aproximadamente 1.33 vezes maior. Esta saída é utilizada para possibilitar que sinais de entrada bipolares, com amplitudes maiores do que 5V, possam ser extendidos de modo a se inserirem no intervalo de entrada do ADC. O funcionamento neste modo corresponde a um maior valor de V_{FSR} como se pode observar pela análise da Tabela 3.1.
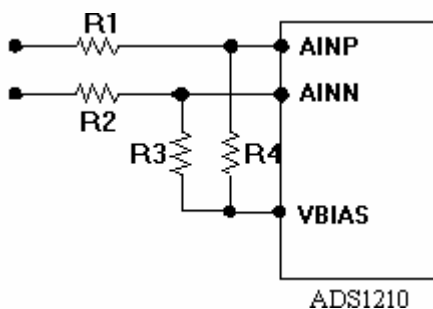
Figura 3.5 – Esquemático do Selector de Modo

Ganho	Entrada Analógicas		Entradas Analógicas utilizando V_{BIAS}	
	V_{FSR} (V)	V_{IN} (V)	V_{FSR} (V)	V_{IN} (V)
1	10	0 a 5	40	±10
2	5	1.25 a 3.75	20	±5
4	2.5	1.88 a 3.13	10	±2.5
8	1.25	2.19 a 2.81	5	±1.25
16	0.625	2.34 a 2.66	2.5	±0.625

Tabela 3.1 – Voltage Full-Scale Range (V_{FSR}) e tensão de entrada (V_{IN}) versus ganho

Os valores das resistências foram criteriosamente escolhidos para que a potência do sinal dissipada seja mínima. Para que isto seja possível as resistências R1 e R2 devem ser três vezes superiores a R3 e R4.

3.3 Implementação da Parte Digital

A parte digital é constituída pelo circuito de lógica programável, pelo buffer, pelo cristal oscilador e pelas fichas de barramento, que permitem fazer o interface do sistema com a arquitectura de controlo.

3.3.1 Barramento

Os sinais que constituem o barramento podem dividir-se em dois grupos, sinais de interface e de endereço.

Dentro do grupo sinais de interface existem:

Serial Data Input - SDIO
Clock Input/Output - SCLK
Serial Data Output – SDOUT
Data Ready - DRDY

O sinal SDIO é utilizado para a escrita de dados provenientes do microcontrolador residente na arquitectura de controlo, para os conversores que constituem o sistema. Esses dados incluem as configurações de cada conversor (ganho, turbo mode, taxa de dados) e a ordem para leitura dos resultados de conversão. A escrita de dados no conversor é feita através do registo *Instruction Register* (INSR) do microcontrolador de cada conversor. Em cada PCI do sistema o sinal passa numa primeira fase por um buffer, indo depois para cada um dos conversores.

O sinal SCLK é um relógio de entrada para a transferência de dados. Devido aos conversores se encontrarem a funcionar em *Slave Mode,* a máxima frequência SCLK não poderá exceder a frequência de relógio (X_{IN}) a dividir por cinco.

SDOUT é o sinal de interface que possibilita a entrega dos resultados de conversão ao microcontrolador da arquitectura de controlo. Cada ADC no fim de uma conversão guarda os seus resultados no registo *Data Output Register* – DOR do seu microcontrolador, tendo-se o período de tempo definido por $1/f_{DATA}-12*(1/f_{XIN})$ para a sua leitura até um novo resultado de conversão se sobrepor ao antigo. Este sinal é o resultado de uma combinação lógica dos sinais SDOUT provenientes de cada conversor

de uma mesma PCI (SDOUT1, SDOUT2, SDOUT3) com o SDOUT proveniente da PCI anterior (SDOUT_ANT).

Por último, DRDY é utilizado para a indicação de resultados de conversão válidos ao microcontrolador da arquitectura de controlo, afim deste poder proceder à sua leitura. Apenas quando este sinal tem a transição para o nível lógico baixo, novos resultados de conversão darão entrada no registo DOR do microcontrolador de cada conversor. DRDY, à semelhança do sinal SDOUT, é uma combinação lógica dos sinais DRDY proveniente do conversor denominado *Master* de cada PCI (DRDYM) e o DRDY da PCI anterior (DRDY_ANT).

Quanto ao segundo grupo, os sinais de endereço que existem são os seguintes:

Linha de endereço 0 – ADDR0
Linha de endereço 1 – ADDR1
Linha de endereço 2 – ADDR2
Linha de endereço 3 – ADDR3

As linhas ADDR0 e ADDR1 são os endereços menos significativos (ADDR0 – linha de menor peso), e que permitem a selecção de cada um dos conversores dentro de cada PCI. As linhas de endereço ADDR2 e ADDR3, os endereços mais significativos (ADDR3 – linha de maior peso), permitem a selecção de cada PCI do sistema. Apesar do carácter diferenciado das linhas de endereço as selecções, não são executadas isoladamente. È através da combinação entre as quatro linhas no circuito de lógica programável que cada PCI e por consequência cada seu conversor pode ou não ser seleccionado.

3.3.2 Buffer

Quando um sinal digital é aplicado à entrada de uma porta deve ser capaz de estabelecer nessa entrada uma ou outra tensão correspondente a um ou outro nível lógico. Em qualquer um dos níveis a fonte deve satisfazer os requisitos de corrente da porta accionada, evitando assim trocas de níveis lógicos. Para que este requisito seja cumprido é necessário conhecer com exactidão o número máximo de entradas que uma porta pode accionar, ou seja o fan-out.

Devido a cada um dos sinais provenientes do microcontrolador serem aplicados a várias entradas do ADC, utilizou-se um buffer para se conseguir satisfazer o requisito acima descrito e assim garantir que não há carga excessiva para a saída do microcontrolador.

Os sinais provenientes do microcontrolador que passam pelo buffer são o SCLK e o SDIO. Estes sinais vão ligar a todas as entradas com o mesmo nome dos ADC, como se representa na figura seguinte.
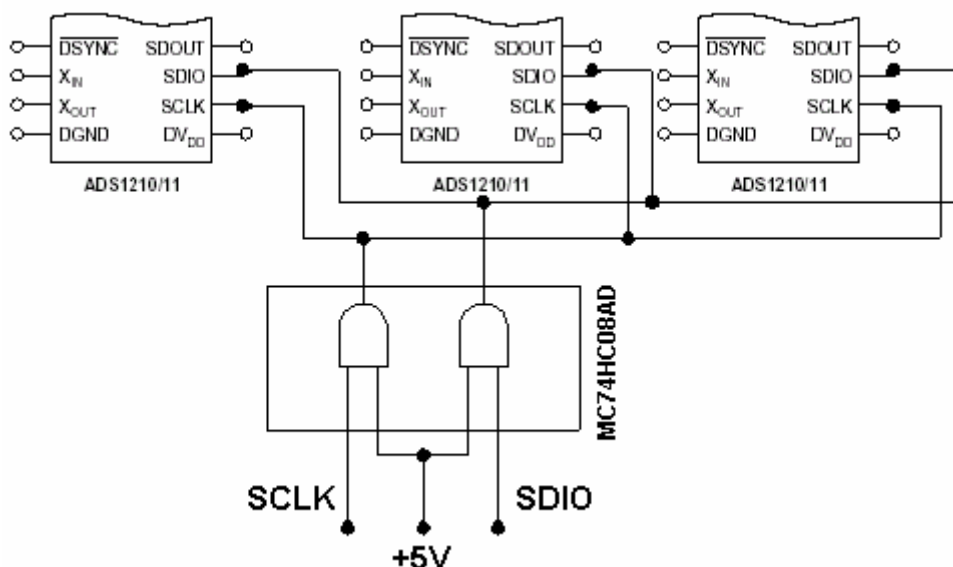
Figura 3.6 – Diagrama de Bloco de ligações do buffer em cada PAS

3.3.3 Lógica programável

Este bloco é muito importante para o correcto funcionamento de todo o sistema. Está dividido em três partes distintas que são activação/desactivação do ADC, sinais de SDOUT e de DRDY e por último, o circuito de sincronização.

Este bloco é concretizado por uma PAL P5Z22V10. As suas características principais podem ser consultadas no anexo C. A programação da mesma também se encontra no anexo C, e inclui todas as funcionalidades por ela realizada e já citadas.

3.3.4 Activação/Desactivação dos ADC's

Quando existem mais que um dispositivo eléctrico a utilizar o mesmo barramento de saída de sinais, em que a transferência de dados pode ocorrer ao mesmo tempo, há a necessidade de seleccionar o dispositivo para se identificar de onde é que os dados são provenientes.

Para tal, no caso do presente trabalho, usa-se o sinal de CS para controlar o estado "*tri-state*"das saídas dos ADC. Este sinal quando colocado a um lógico, coloca a saída do ADC no estado "*tri-state*".

O sinal CS é o resultado de uma combinação lógica dos sinais de endereço, que é realizada pela PAL. Em primeiro lugar é necessário seleccionar a PAS onde se encontra o ADC. Cada PAS é numerada de zero a três através dos *jumpers* J1 e J2. Com a ajuda de um de comparador, compara-se o valor da PAS com os sinais mais significativos de endereço, seleccionando assim a PAS, como a seguir se exemplifica.

J1	J2	ADDR3	ADDR2	
0	0	0	0	PAS1
0	1	0	1	PAS2
1	0	1	0	PAS3
1	1	1	1	PAS4

Tabela 3.2 – Selecção da PAS

Após se seleccionar a PAS onde se encontra o ADC pretendido, tem que se seleccionar o ADC. Para isso utiliza-se os dois sinais de endereçamento que restam, ou seja, os menos significativos. Na tabela seguinte indica-se as combinações destes dois sinais que seleccionam o ADC dentro da PAS.

ADDR1	ADDR0	
0	0	ADC1
0	1	ADC2
1	0	ADC3
1	1	Casos específicos

Tabela 3.3 – Selecção do ADC em cada PAS

O circuito lógico que a PAL realiza para a activação/desactivação do ADC é o seguinte
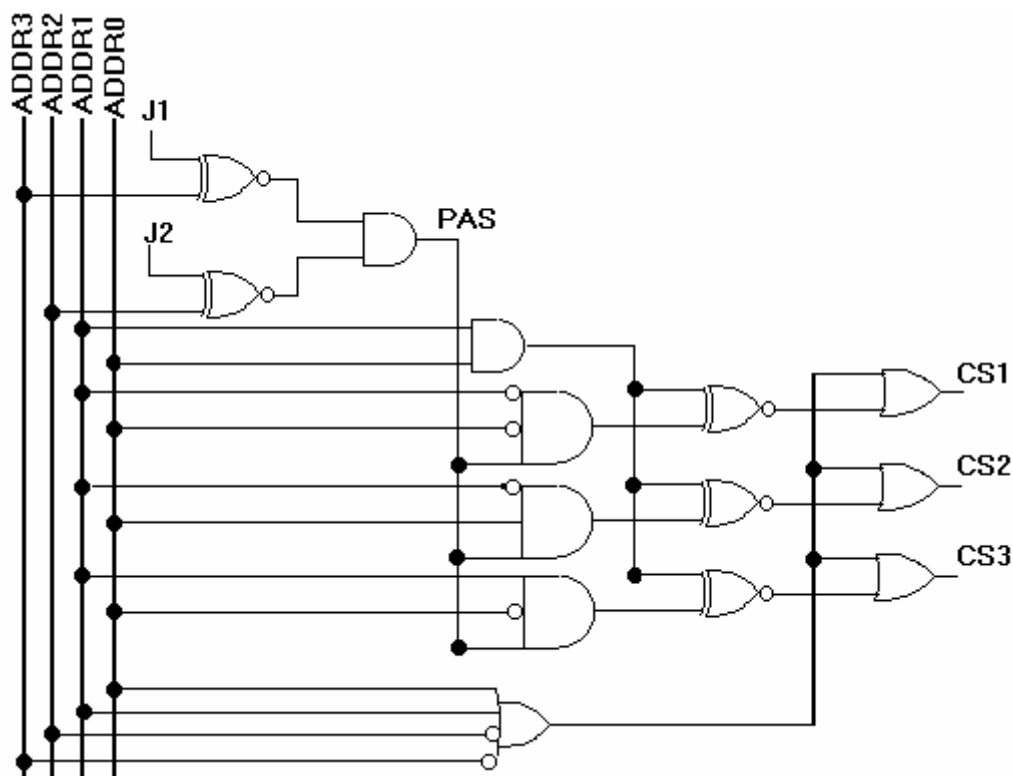
Figura 3.7 – Esquema lógico que permite a selecção de cada ADC

Com este circuito podemos seleccionar apenas um ADC de cada vez, isto é, o ADC seleccionado fica com o sinal CS a zero lógico e os restantes a um lógico.

ADDR3	ADDR2	ADDR1	ADDR0	Função
0	0	1	1	Coloca todos os ADC com o CS a 1
0	1	1	1	Sincroniza todos os ADC
1	0	1	1	Sem função
1	1	1	1	Coloca todos os ADC com o CS a 0
Outros casos				Coloca o CS a 1do ADC especificado

Tabela 3.4 – Tabela de funções especificas

3.3.5 Sinais de DRDY e SDOUT

O DRDY é um sinal gerado pelo ADC para indicar o fim de uma conversão e portanto indica que os dados já podem ser lidos. Em cada PAS apenas é aproveitado um destes sinais, que se denomina de DRDYM e o ADC que o gerou denomina-se de ADC "master".

O microcontrolador necessita saber quando é que pode ler os dados resultantes de uma conversão. A indicação da situação que o microcontrolador necessita, é obtida pela combinação dos sinais de DRDYM gerados pelo ADC "master" de cada PAS.

Para que chegue ao microcontrolador apenas um sinal deste tipo é necessário que em cada PAS seja feita uma combinação do sinal DRDYM com o sinal de DRDY da PAS anterior, de tal modo que quando se chega à PAS que se encontra ligado ao microcontrolador, o sinal resultante é o sinal que indica o fim de conversão.

Os sinais combinam-se da seguinte forma
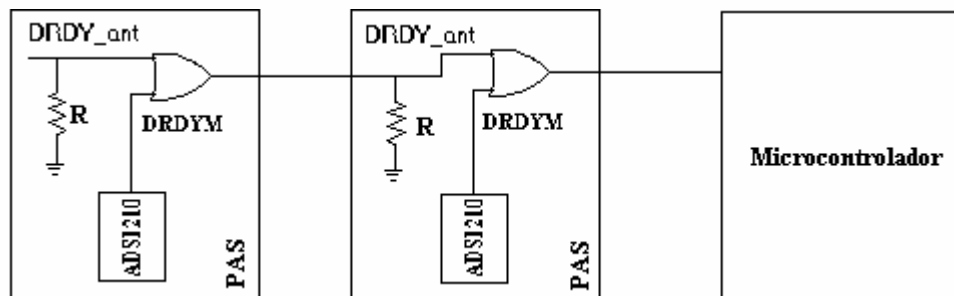
Figura 3.8 – Selecção do DRDY de entrada no microcontrolador

O DRDY_ANT é o DRDY da PAS anterior. Cada entrada de DRDY_ANT possui um *pull-down* para o caso dessa PAS ser a última. Assim nessa PAS o sinal DRDY_ANT é zero lógico e não interfere no resultado final.

O sinal SDOUT é o sinal que entrega ao microcontrolador o resultado de uma conversão. Como é fácil de perceber é importante que o microcontrolador possa determinar a origem dos dados que quer receber.

No sistema pode existir várias PAS e é sabido que em cada PAS existe três sinais do tipo SDOUT, o SDOUT1, SDOUT2 e o SDOUT3. Então o microcontrolador deve primeiro seleccionar o ADC desejado, obrigando deste modo a que as saídas dos restantes ADC's fiquem no estado "*tri-state*". Desta forma e com a ajuda dos sinais de CS o microcontrolador recebe dados apenas do ADC pretendido.

O SDOUT que chega ao microcontrolador é oriundo do seguinte circuito lógico implementado na PAL.
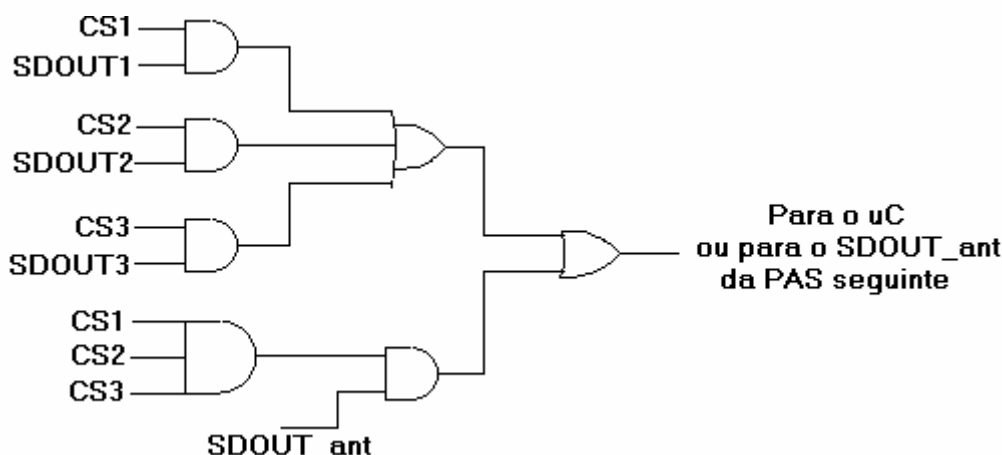
Figura 3.9 – Esquema lógico que permite selecção de SDOUT

O sinal SDOUT_ANT é uma entrada para os dados oriundos das PAS anteriores. Pelas mesmas razões apresentadas anteriormente, o sinal SDOUT_ANT também possui um *pull-down*.

3.3.6 Circuito de Sincronização

Nos ADC's a sincronização é feita por um pino do ADC denominado DSYNC activo na transição de zero para um lógico.

A necessidade desta operação é para que todos os ADC do sistema comecem a converter no mesmo instante. Desta forma o sinal de DRDY é igual para todos os ADC e assim justifica-se o uso de apenas um, o DRDYM, sem ocorrerem problemas.

No diagrama abaixo representa-se a situação transcrita acima.
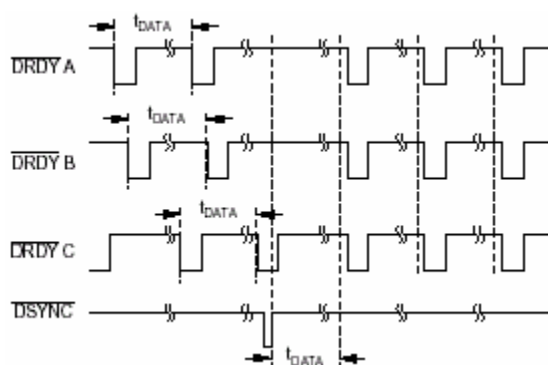
Figura 3.10 – Diagrama Temporal de uma situação onde é efectuado o sincronismo

Esta situação só é conseguido à custa de um circuito lógico implementado pela PAL e por um circuito ressonante, como de seguida se representa.
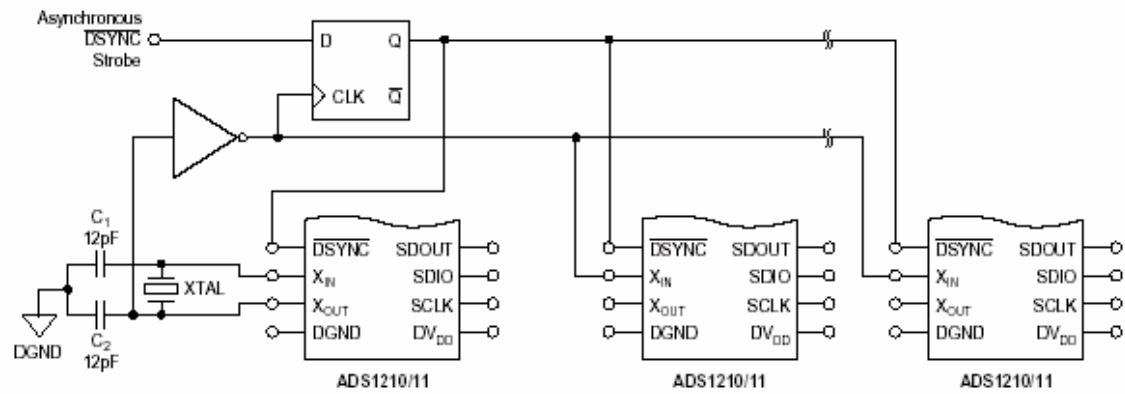
Figura 3.11 – Diagrama de Blocos das ligações sincronismo em cada PAS

Com este circuito garante-se que o sinal assíncrono DSYNC *strobe* na sua transição de zero para um lógico, tome o valor de um lógico para todos os ADC no flanco descendente do sinal de relógio, constituído pelo cristal. Assim garante-se que todos os ADC comecem a converter no mesmo instante.

O sinal de DSYNC *strobe* é conseguido à custa dos sinais de endereço, com a combinação da tabela abaixo, a qual permite sincronizar todos os ADC's existentes no sistema.

ADDR3	ADDR2	ADDR1	ADDR0	DSYNC strobe
0	1	1	1	0

Tabela 3.5 – Endereço específico para se efectuar o sincronismo

4 Controlo de Temperatura dos Sensores

4.1 Introdução

Após a elaboração da arquitectura de aquisição de sinais, explicada no capítulo anterior, é vantajoso estabilizar a temperatura dos sensores que fornecem os sinais analógicos afins de serem convertidos para sinais digitais. Só assim se podem obter leituras com elevado grau de fiabilidade e independentes da temperatura de trabalho.

O grande problema dos sensores é a variação das medidas tomadas com a temperatura, isto é, para uma determinada temperatura e condições de funcionamento os sensores dão uma determinada medida, enquanto para outra temperatura diferente e para as mesmas condições os sensores dão uma medida diferente, o que obviamente é de todo indesejável, devido ao erro de leituras que introduz. Um exemplo simples deste incómodo problema, é por exemplo, a situação em que se tem um veículo parado e um acelerómetro solidário com o mesmo indicar que se encontra com uma determinada aceleração, a qual pode variar se a temperatura do sensor variar.

Para a resolução desse problema existem duas soluções possíveis: a primeira implica o conhecimento do erro de polarização em relação a cada ponto da temperatura de funcionamento dos sensores, de forma a permitir saber em cada instante de tempo, a compensação a efectuar nas grandezas medidas; a segunda corresponde à estabilização da temperatura de funcionamento dos sensores, encontrando-se todas as medidas afectadas do mesmo valor. A solução adoptada foi a segunda devido a ser a mais fácil de implementar e requerer uma menor quantidade de recursos, tanto técnicos como científicos.

4.2 Arquitectura de Controlo de Temperatura

Para se conseguir efectuar o controlo de temperatura foi indispensável escolher um sensor de temperatura. Com esse intuito escolheu-se uma resistência, denominada PRC-100, com um coeficiente de temperatura de 0.385Ω /ºC, o qual pode-se assumir constante para a gama de interesse.

A temperatura de funcionamento dos sensores foi outro ponto importante na construção desta arquitectura. Esta temperatura não podia ser a temperatura ambiente, devido a esta variar ao longo do tempo, nem uma temperatura muito elevada que requer o gasto de muita energia, recurso escasso a bordo do helicóptero autónomo. Assim, seleccionou-se uma temperatura intermédia de 50ºC, para temperatura de trabalho. Para se conseguir esta temperatura tem que se recorrer a uma fonte de calor inserida num meio fechado, de forma a minimizar o fluxo de calor para o exterior. A forma mais simples de se conseguirem estes dois factores foi a construção de uma caixa com material de baixa condutibilidade térmica e fácil de trabalhar. Entre o leque de materiais estudados, optou-se pelo poliestireno extrudido devido a apresentar as melhores características, menor valor de condutibilidade térmica ($0.027 Wm^{-1}K^{-1}$), ser mais fácil e rápido de trabalhar, e se apresentar como o mais barato. Dentro dessa caixa inseriu-se outra caixa, esta de metal, de modo ao fluxo de calor se propagar de uma forma homogénea por todas as suas faces, bem como no seu interior. Por sua vez, no interior desta caixa colocou-se o sistema de aquecimento formado por quatro resistências de

potência, os sensores e um sistema de controlo que efectua a estabilização da temperatura dos sensores.

Para se conseguir estabilizar a temperatura a 50ºC foi necessário ter em conta as trocas de temperatura com o meio exterior, que apesar de se tentarem ser mínimas, existem. Assim foi efectuado o cálculo da potência que será necessário colocar ao longo do tempo dentro da caixa interior, de modo a ter sempre uma temperatura estabilizada na ordem dos 50ºC.
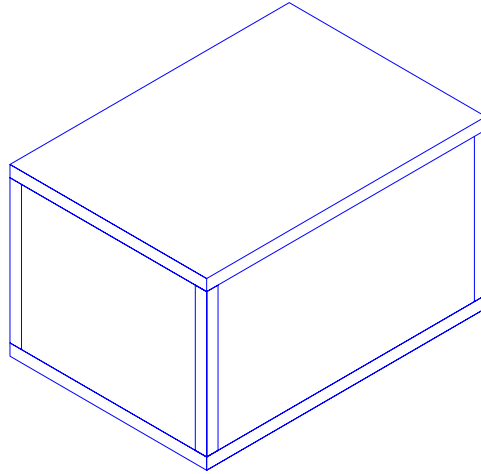
Figura 4.1 – Perspectiva da Caixa Exterior totalmente fechada

Figura 4.2 – Perspectiva do Interior das Caixas, Exterior e Interior

O cálculo da Potência Fornecida (P – Watt) pelo sistema é feito de acordo com a expressão seguinte:

$$P = \frac{U^2}{R_L} \qquad (4.1)$$

onde

R_L – Valor Total da Resistência de Potência (resistência equivalente do sistema de aquecimento)
U – Tensão aos terminais de R_L

O fluxo de calor que se desloca de uma zona quente para uma zona fria, denominada Potência Perdida (ϕ - Watt) pelo sistema, é calculada de acordo com a expressão (4.2).

$$\phi = \frac{\lambda}{S} \times A \times (T - T_A) \qquad (4.2)$$

λ – Condutibilidade térmica
S – Espessura do material que forma a caixa exterior (poliestireno extrudido)
A – Área da caixa exterior
T – Temperatura zona quente
T_A – Temperatura zona fria (temperatura ambiente ≈ 20ºC)

A resposta do sistema é dada pela expressão (4.3) que representa a variação da energia sob a forma de calor no tempo.

$$C\frac{dT}{dt} = \frac{U^2}{R_L} - \frac{A \times \lambda}{S} \times (T - T_A) \qquad (4.3)$$

$$C\frac{dT}{dt} = P - \phi$$

em que
T – Temperatura [K]

$$C = Cesp \times m \qquad (4.4)$$

na qual
Cesp – Calor especifico $[JKg^{-1}K^{-1}] = 0.9 \times 10^3 JKg^{-1}K^{-1}$
m – massa do sensor [Kg] = 0.1Kg

Inicialmente a potência fornecida para aquecer o sistema, P, será superior ao valor da potência perdida, ϕ, chegando ao ponto de equilíbrio quando se atingir a temperatura de 50ºC. Note-se que de maneira alguma a potência perdida pode ser superior à potência fornecida. Em equilíbrio a potência a ser fornecida é igual à potência perdida pelo sistema.

O ponto de equilíbrio do sistema corresponde a ter a potência perdida totalmente compensada pela potência fornecida, isto é, $\phi = P$.

Quando o sistema se encontra no seu ponto de equilíbrio, isso equivale a dizer que a variação de energia sob a forma de calor no tempo, que é traduzida pela expressão (4.3) é igual a zero, pois como se viu nesse ponto $\phi = P$.

Considerando,
T = 50ºC
T_A = 20ºC
Dimensões da caixa exterior (90×66×51)mm
$\lambda = 0.027 Wm^{-1}K^{-1}$
S = 40mm

de acordo com a expressão (4.2), ϕ = 0.5628 W, pelo que para se ter o equilíbrio, $C\dfrac{dT}{dt} = 0$, tem-se P = ϕ = 0.5628 W.

4.3 Esquema Eléctrico

Neste ponto é apresentado o esquema eléctrico da arquitectura de controlo de temperatura. Este esquema é um circuito simples, constituído por um amplificador operacional, oito resistências de grande estabilidade térmica, um condensador, um díodo, uma resistência de variação com a temperatura e uma carga, que é o sistema de aquecimento.
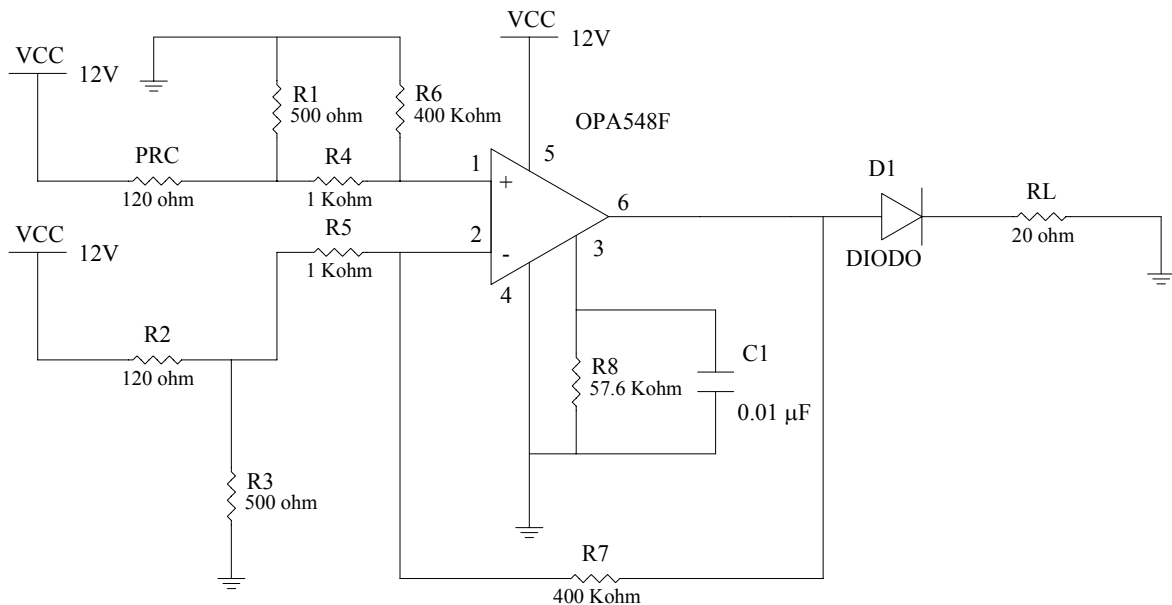
Figura 4.3 – Esquema Eléctrico da Arquitectura de Controlo de Temperatura

A resistência R1 e a resistência PRC-100 formam um divisor de tensão, cuja sua tensão equivalente varia com a temperatura. Sempre que a temperatura for igual ou superior à temperatura desejada o sistema de aquecimento estará desligado; caso a temperatura começa a diminuir o sistema de aquecimento começará a debitar potência e a aquecer o ambiente interno da caixa.

A resistência R2 e R3 formam outro divisor de tensão, que é utilizado como referência, o qual indica o valor de tensão correspondente à temperatura desejada, neste caso 50ºC.

Quando a temperatura desce a diferença de tensão aos terminais do amplificador aumenta, aumentando o valor da tensão de saída e o sistema de aquecimento é ligado.

A resistência R8 é utilizada para limitar a intensidade de corrente de saída do amplificador operacional a 1A. O condensador C1, em paralelo com essa resistência é utilizado para obviar problemas de ruído.

O díodo colocado entre a saída do amplificador e a carga, sistema de aquecimento, tem a função de apenas permitir que esta seja alimentada por um valor positivo de tensão.

O sistema de aquecimento com uma resistência equivalente de 20Ω é constituído por quatro resistências de potência de 5 Ω, espalhadas pelas paredes da caixa interior, exceptuando a parede de fundo e de cima.

4.4 Diagrama de Blocos

A figura 4.4 ilustra o diagrama de blocos que representa a arquitectura de controlo de temperatura.
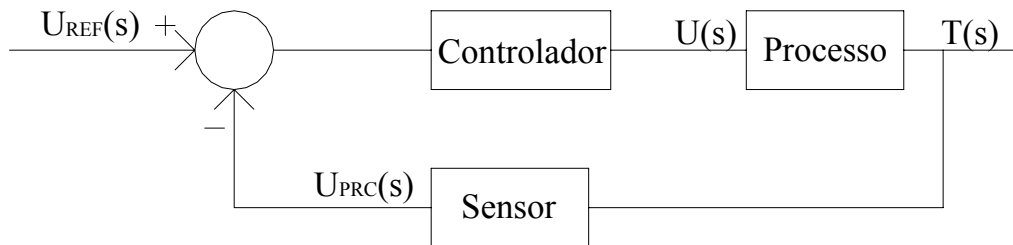
Figura 4.4 – Diagrama de Blocos

Por ser simples de implementar escolheu-se um controlador proporcional. O controlador toma conhecimento da diferença de temperatura (temperatura de saída e temperatura desejada), expressa em termos de um sinal eléctrico (tensão), e actua sobre o processo injectando-lhe um sinal de comando. O processo é o bloco que representa o comportamento do sistema. Consoante o sinal na sua entrada, ele converterá esse sinal numa temperatura maior ou menor. O sensor é o bloco que permite ter acesso à temperatura de saída, T(s).

O sistema tem como sinal de entrada um valor de tensão, denominada tensão de referência $U_{REF}(s)$, equivalente à temperatura que se quer que o sistema funcione, temperatura desejada. A sua saída é o sinal T(s), que significa a temperatura efectiva do sistema, isto é, a temperatura a que efectivamente o sistema está a operar, e é o sinal que se pretende controlar.

Cada bloco é caracterizado pela sua função de transferência que seguidamente será apresentada sucintamente:

<u>Controlador</u>

K – Ganho do sistema

$$K = \frac{R7}{R5} \qquad (4.5)$$

onde R7 = 400KΩ e R5 = 1KΩ, resultando K = 400.

Processo

Este bloco tem como entrada um sinal eléctrico, tensão de saída do amplificador operacional, representada como o sinal U(s), que converte numa temperatura que é colocada à sua saída, T(s). A função transferência deste bloco é obtida a partir da expressão (4.3) e é definida como a relação da transformada de Laplace da saída com a transformada de Laplace da entrada, considerando nulas todas as condições iniciais.

Sinal de Entrada – U
Sinal de Saída – T

Aplicando a fórmula de Taylor com aproximação de 1ª ordem, à expressão (4.1) consegue-se obter uma linearização da tensão de saída em torno do seu ponto de equilíbrio (u_0=3.355V, T_0=50ºC) no qual se tem $\phi = P$, como se demonstra:

$$\mathrm{f}(\mathrm{u}) \approx f(u_0) + f'(u_0)\delta u$$

$$\frac{u^2}{R} \approx \frac{u_0^2}{R} + \frac{2u_0}{R}\delta u$$

Substituindo na expressão (4.3) $T = T_0 + \delta T$ e $u = u_0 + \delta u$ obtêm-se:

$$C\frac{d}{dt}(T_0 + \delta T) = \frac{u_0^2}{R} + \frac{2u_0}{R}\delta u - \frac{A\lambda}{S}(T_0 - T_A) - \frac{A\lambda}{S}\delta T,$$

simplificando

$$C\frac{d}{dt}\delta T = \frac{2u_0}{R}\delta u - \frac{A\lambda}{S}\delta T$$

Aplicando a Transformada de Laplace à equação obtêm-se:

$$s\delta T(s) = b\delta u(s) - a\delta T(s)$$

onde $b = \frac{2u_0}{CR}$, $a = \frac{A\lambda}{SC}$, R = 20 Ω e C é dado pela expressão (4.4).

Então a função transferência é a seguinte expressão:

$$\frac{\delta T(s)}{\delta u(s)} = \frac{b}{s+a} \qquad (4.6)$$

Substituindo as várias variáveis pelos seus valores numéricos na expressão (4.6), obtêm-se:

$$\frac{\delta T(s)}{\delta u(s)} = \frac{3.728\times10^{-3}}{s + 2.08\times10^{-4}}$$

Sensor

Este bloco tem a função de medir a temperatura de funcionamento do sistema, tendo como entrada essa mesma temperatura, δT(s), que converte num sinal eléctrico, um valor de tensão, U_{PRC}, para poder ser comparado com a tensão de referência, que indica a temperatura desejada e assim se verificar se o sistema de aquecimento deve aquecer ou não o ambiente onde estão inseridos os sensores. O bloco tem como sinal de entrada a temperatura efectiva do sistema, δT(s), e como sinal de saída o seu valor convertido para tensão eléctrica, denominado $U_{PRC}(s)$. A função transferência é obtida a partir da expressão (4.7), que traduz tensão de saída do amplificador, δu, em função de U_{REF} e U_{PRC}.

$$\delta u = K \times (U_{PRC} - U_{REF}) \qquad (4.7)$$

onde K é o ganho do sistema, U_{PRC} é dado pela expressão (4.8) e U_{REF} pela expressão (4.9).

$$U_{PRC} = \frac{R1}{R1 + PRC} Vcc \qquad (4.8)$$

$$U_{REF} = \frac{R3}{R2 + R3} Vcc \qquad (4.9)$$

Na expressão (4.8) PRC é substituído pela expressão (4.10), que traduz a sua variação com a temperatura:

$$PRC = Ro + \alpha \times \delta T \qquad (4.10)$$

na qual Ro é o valor da resistência a 0ºC (100Ω), α o seu coeficiente de temperatura e δT a temperatura (temperatura a que está a funcionar o sistema). Nas expressões (4.8) e (4.9) a variável Vcc representa a tensão de alimentação do sistema que é de 12V.

Substituindo as variáveis U_{PRC}, U_{REF} e K na expressão (4.7) por valores numéricos, obtêm-se uma expressão que traduz δu em função da temperatura a que opera o sistema.

$$\delta u = 400 \times \left(\frac{500}{500 + 100 + \alpha \times \delta T} Vcc - \frac{500}{500 + 120} Vcc \right)$$

$$\delta u = 400 \times \left(\frac{500 \times 620 - 500 \times (500 + 100 + \alpha \times \delta T)}{620 \times (500 + \underline{100 + \alpha \times \delta T})} \right) \qquad (4.11)$$

↓
em torno da temperatura
de referência 50ºC→120 Ω

Simplificando a expressão (4.11):

$$\delta u = 400 \times \left(0.026Vcc - \frac{500\alpha \times \delta T}{620^2} Vcc \right)$$

onde 400 é o valor do controlador, 0.026Vcc = U_{REF} e $\frac{500\alpha T}{620^2} Vcc$ = U_{PRC}, logo a função de transferência deste bloco é a seguinte:

$$\frac{U_{PRC}(s)}{\delta T(s)} = 1.3 \times 10^{-3} \alpha Vcc$$

A função de transferência do sistema em cadeia fechada é então definida por:

$$\frac{\delta T(s)}{U_{REF}(s)} = \frac{Controlador \times \Pr ocesso}{1 + Controlador \times \Pr ocesso \times Sensor} \qquad (4.12)$$

substituindo pela função de transferência de cada bloco, tem-se:

$$\frac{\delta T(s)}{U_{REF}(s)} = \frac{1.49}{s + 9.2 \times 10^{-3}}$$

resultando num pólo em s= -9.2×10^{-3}.

Para averiguar acerca da estabilidade do sistema foi efectuado o seu estudo, quanto à localização dos seus pólos e a sua resposta na frequência (através do seu Diagrama de Bode – Características de Amplitude).
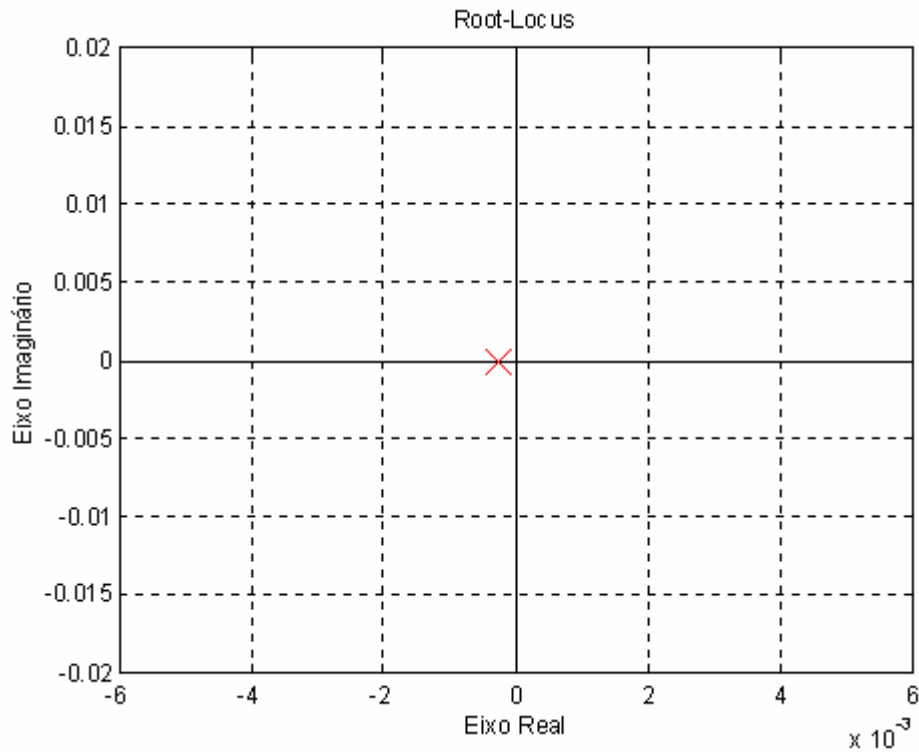
Figura 4.5 – Mapa dos Pólos de Função Transferência em cadeia aberta
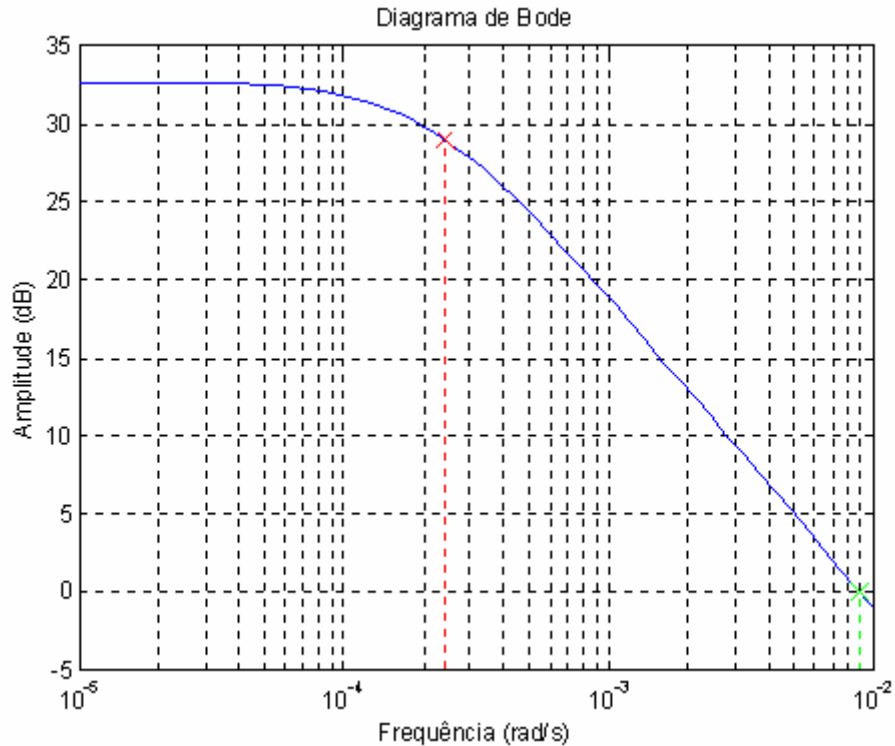
Figura 4.6 – Diagrama de Bode de Função Transferência em cadeia aberta

Como se pode verificar pela análise da Figura 4.5 e em comparação com o pólo calculado para cadeia fechada, verifica-se que apesar do sistema em ambas as situações ser um sistema estável (pólos no semi-plano complexo esquerdo), o sistema em cadeia fechada tem uma maior rapidez de resposta no seguimento da sua referência. A partir de uma análise da Figura 4.6 pode-se constatar isso mesmo. Com a cor vermelha encontra--se marcada a largura de banda do sistema em malha aberta; com tracejado a verde assinala-se a largura de banda do sistema em malha fechada. Contudo para ambos os casos o sistema apresenta uma resposta lenta.

4.5 Placa de Circuito Impresso

Após um elaborado estudo foi elaborada uma placa de circuito impresso, afim de ser possível a implementação da Arquitectura de Controlo de Temperatura. Nas figuras seguintes encontra-se representado a placa completa, a sua face superior (*top*), a sua face inferior (*bot*), assim como as suas respectivas máscaras de protecção de pistas (*mask*). As dimensões das figuras não apresentam nenhuma relação com as dimensões da placa original. As dimensões desta são 3.3×4.4 cm

Figura 4.7 – Placa Completa

Figura 4.8 – Face Superior

Figura 4.9 – Máscara da Face Superior

Figura 4.10 – Face Inferior

Figura 4.11 – Máscara da Face Inferior

5 Conclusão

Os objectivos deste trabalho, apresentados na introdução, têm como parâmetro principal de qualidade a obtenção da resolução efectiva mais elevada possível e deste modo explorar ao máximo todas as potencialidades do conversor analógico-digital.

Foram efectuadas duas iterações da placa que implementa a arquitectura de aquisição de dados, com vista a obter uma resolução cada vez melhor. Neste momento conseguiu-se 16 bits, o que se considera já razoável quando comparada com sistemas de aquisição existentes no mercado. Com a intenção de obter uma resolução efectiva mais alta foi elaborada uma nova placa de circuito impresso, de várias camadas fabricada no estrangeiro, que chegou muito próximo do prazo de entrega do relatório, não sendo possível executar os devidos testes.

O circuito electrónico para o sistema de controlo de temperatura de sensores foi testado numa *breadboard* estando a quando da finalização deste relatório a placa de circuito impresso pronta para os testes finais.

Os resultados finais do trabalho serão entregues o mais brevemente possível, numa adenda a este relatório. Como resultados finais inclui-se a resolução da placa de aquisição e os testes do controlador de temperatura já com a placa final.

Foi também no âmbito deste trabalho desenvolvido um *driver* utilizando as linguagens C e Assembler que controla a transferência de dados entre o microcontrolador e a placa de aquisição de dados o qual se encontra a funcionar correctamente.

A Manual de utilização da Placa de Aquisição de Sinais

A.1 Descrição

O sistema desenvolvido faz a conversão de sinais analógicos para sinais digitais e fornece-os ao microcontrolador XA-S3, que controla toda a operação de conversão, residente na placa da arquitectura de controlo designada de MC-XAS3.

Como se pretendem dados relativos a diferentes grandezas, o sistema desenvolvido é constituído por quatro placas de circuito impresso completamente iguais, uma por cada grandeza, e como se pretendem dados em relação aos três eixos cartesianos, cada uma das PASs desenvolvidas inclui três conversores. No entanto, deseja-se que os resultados de conversão fornecidos ao microcontrolador sejam referenciados ao mesmo instante de tempo. Para isso a PAS tem todos os seus conversores sincronizados, isto é, o sinal que indica resultados de conversão válidos (DRDY) está no mesmo estado lógico no mesmo instante para cada um dos conversores. A sincronização é feita em cada uma das PASs (este sinal não é passado através do barramento de PAS em PAS), para obviar o efeito de introdução de ruído no sinal de relógio caso este passa-se de PAS em PAS. È através de uma combinação especifica das quatro linhas de endereço do barramento que se executa a sincronização obrigando o pino DSYNC de cada conversor passar do seu estado normal de funcionamento (estado lógico um) para o estado lógico zero, regressando novamente ao estado lógico um e aí permanecendo para qualquer outra combinação das mesmas linhas. Para se obter uma sincronização perfeita em cada PAS, o sinal DSYNC deverá estar sincronizado com o sinal de relógio, isto é, o sinal DSYNC será a saída de um flip--flop existente na Pal, tendo como entrada a combinação das quatro linhas de endereço, activada no flanco descendente de cada ciclo de relógio. Esta aplicação leva à existência de um conversor denominado de *MASTER* em cada uma das PASs do sistema (o conversor U2, conforme esquemático apresentado em figura A.1).

Outra das aplicações do sistema é a possibilidade de se permitir, ou inibir a comunicação com cada um dos conversores. Esta aplicação, e como já foi referido no Capítulo 2 deste relatório, implica que os conversores se encontrem a operar em modo *Slave Mode*, de modo a permitir o controlo do pino CS.

A interface entre o sistema e o microcontrolador que controla toda a operação do mesmo é feita através dos diferentes Portos de Comunicação:

- P6: Interface entre a arquitectura de controlo e o sistema, quando pertencendo à PAS *MASTER*; em todas as outras PASs do sistema é utilizado para receber os sinais de controlo provenientes da PAS anterior.

Porto	Função	MC-XAS3-(P1)	Observações
P 6.1	SDOUT	6	Resultados de Conversão
P 6.2	DRDY	7	Indicação de dados de conversão válidos
P 6.3	SCLK	8	Relógio Entrada/Saída para transferência de dados
P 6.4	SDIO	9	Entrada de dados
P 6.5	GROUND	10	Massa Digital
P 6.6	ADDR3	16	Linhas de endereço/Selecção de PAS
P 6.7	ADDR2	17	Linhas de endereço/Selecção de PAS
P 6.8	ADDR1	18	Linhas de endereço/Selecção de conversor
P 6.9	ADDR0	19	Linhas de endereço/Selecção de conversor

Tabela A.1 – Função do Porto P6

- P7: Este porto é utilizado para interface dos sinais que permitem controlar a operação do conversor de uma placa para outra.

Porto	Função	Porto P 6	Observações
P 7.1	SDOUT_ANT		Resultados de Conversão
P 7.2	DRDY_ANT		Indicação de dados de conversão válidos
P 7.3	SCLK	P 6.3	Relógio Entrada/Saída para transferência de dados
P 7.4	SDIO	P 6.4	Dados de entrada
P 7.5	CGND	P 6.5	Massa da placa da arquitectura de controlo
P 7.6	ADDR3	P 6.6	Linhas de endereço
P 7.7	ADDR2	P 6.7	Linhas de endereço
P 7.8	ADDR1	P 6.8	Linhas de endereço
P 7.9	ADDR0	P 6.9	Linhas de endereço

Tabela A.2 – Função do Porto P7

A massa proveniente da placa da arquitectura de controlo com a qual o porto P6 da primeira PAS do sistema se encontra ligada é passada pelo barramento de modo a se fazer o acoplamento entre as duas massas, DGND (massa digital de cada PAS) e CGND (massa proveniente da placa da arquitectura de controlo). Recomenda-se que o acoplamento de massas seja sempre feito, ou seja, o ponto de acoplamento esteja sempre fechado, de forma se ter uma única referência na parte digital do sistema.

Para se poder seleccionar cada um dos diferentes conversores constituintes do sistema, cada um tem de ter um endereço único. Esse endereço é constituído por quatro bits, os dois mais significativos (ADDR3 e ADDR2) referentes à placa a que pertence e os dois menos significativos (ADDR1 e ADDR0) referentes ao endereço de cada conversor dentro de cada placa. Para se conseguir efectuar esta selecção é necessário um comparador, que mais não é do que o endereço de cada placa. Este comparador, efectuado através de dois *jumpers*. Existe ainda um conjunto de endereços especiais que serão aceites em todas as PASs, os quais se encontram devidamente explicados no Capítulo 3.

J1	P5Z22V10 (PAL)	Observações
2 – 1	3	Valor lógico "1"
2 – 3	3	Valor lógico "0"

Tabela A.3 – Ligações de J1

J2	P5Z22V10 (PAL)	Observações
2 – 1	10	Valor lógico "0"
2 – 3	10	Valor lógico "1"

Tabela A.4 – Ligações de J2

Estes dois *jumpers* permitem a selecção de um endereço único para cada PAS do sistema. Como o sistema é constituído por quatro PAS, 2 bits são suficientes (daí ser necessário apenas dois *jumpers*) para a selecção de um endereço por PAS. A tabela com os endereços possíveis de seleccionar com estes dois *jumpers* encontra-se apresentada no capítulo 3.3.4.

Existe ainda um outro conjunto de *jumpers*, com a função de permitir alterar o modo de operação de cada um dos conversores, modo comum para modo simétrico, ou vice-versa, podendo um conversor estar a funcionar num modo e um outro qualquer dentro da mesma PAS, no outro modo. Para operar em cada um destes modos, vasta apenas colocar os respectivos *jumpers*.

Jumper	Modo de Operação
JP1, JP2, JP3, JP4, JP5, JP6	Modo Comum
JP7, JP8, JP9	Modo Simétrico

Tabela A.5 – Modos de Operação

A.2 Esquema Eléctrico

Apresenta-se na Figura A.1 o esquema eléctrico da placa desenvolvida.
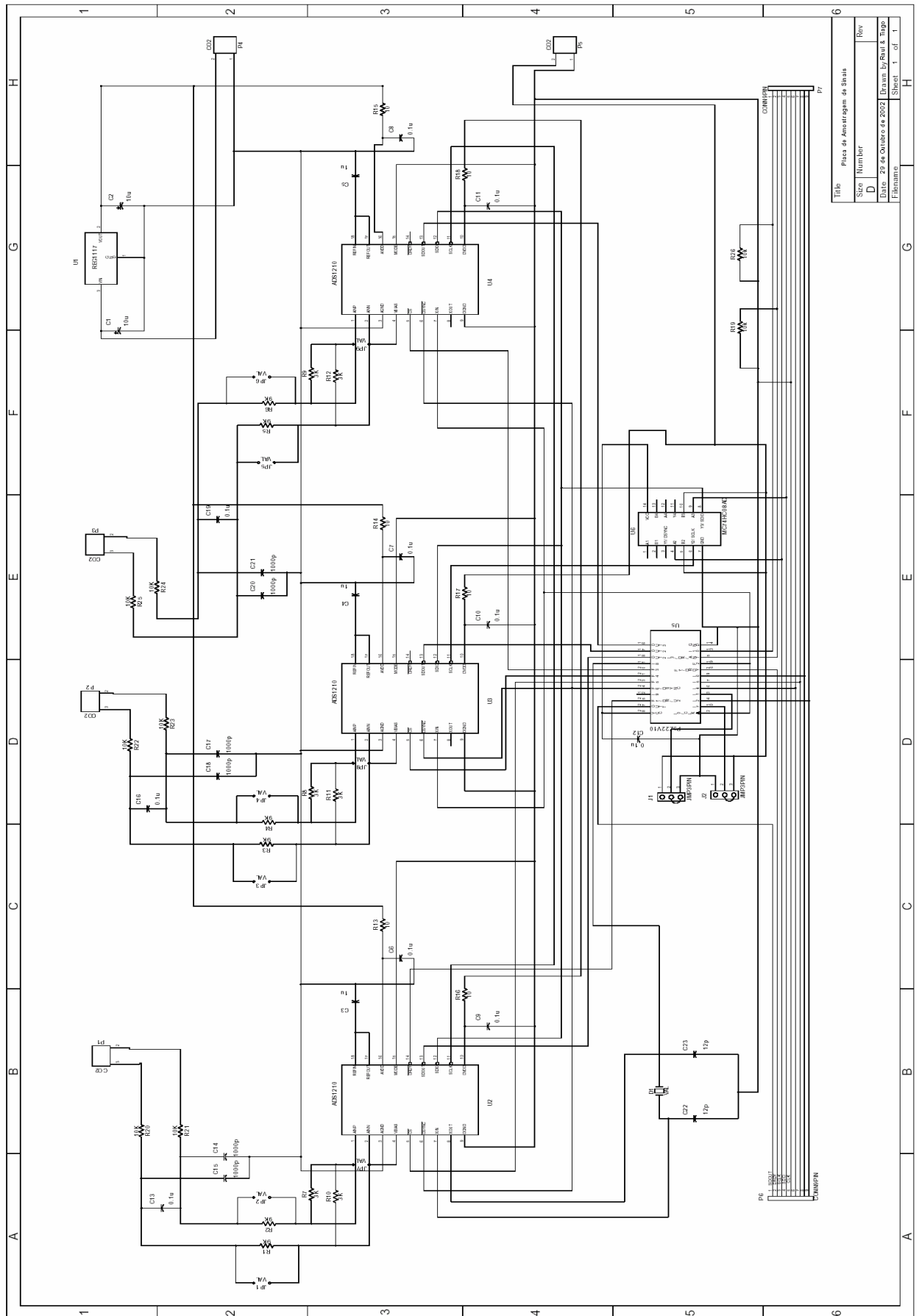
Figura A.1 – Esquema Eléctrico da Placa Aquisição de Sinais

A.3 Placa de Circuito Impresso

Com base no esquema eléctrico anterior foi projectada uma PAS de duas faces com as dimensões de 4.5×5.5cm. Nas figuras A.2 e A.3 apresenta-se a face superior e a face inferior da referida placa, assim como as máscaras de protecção de suas pistas e suas marcações. Na figura A.4 apresenta-se o PAS da placa completa. Nenhuma escala é utilizada na apresentação das figuras.
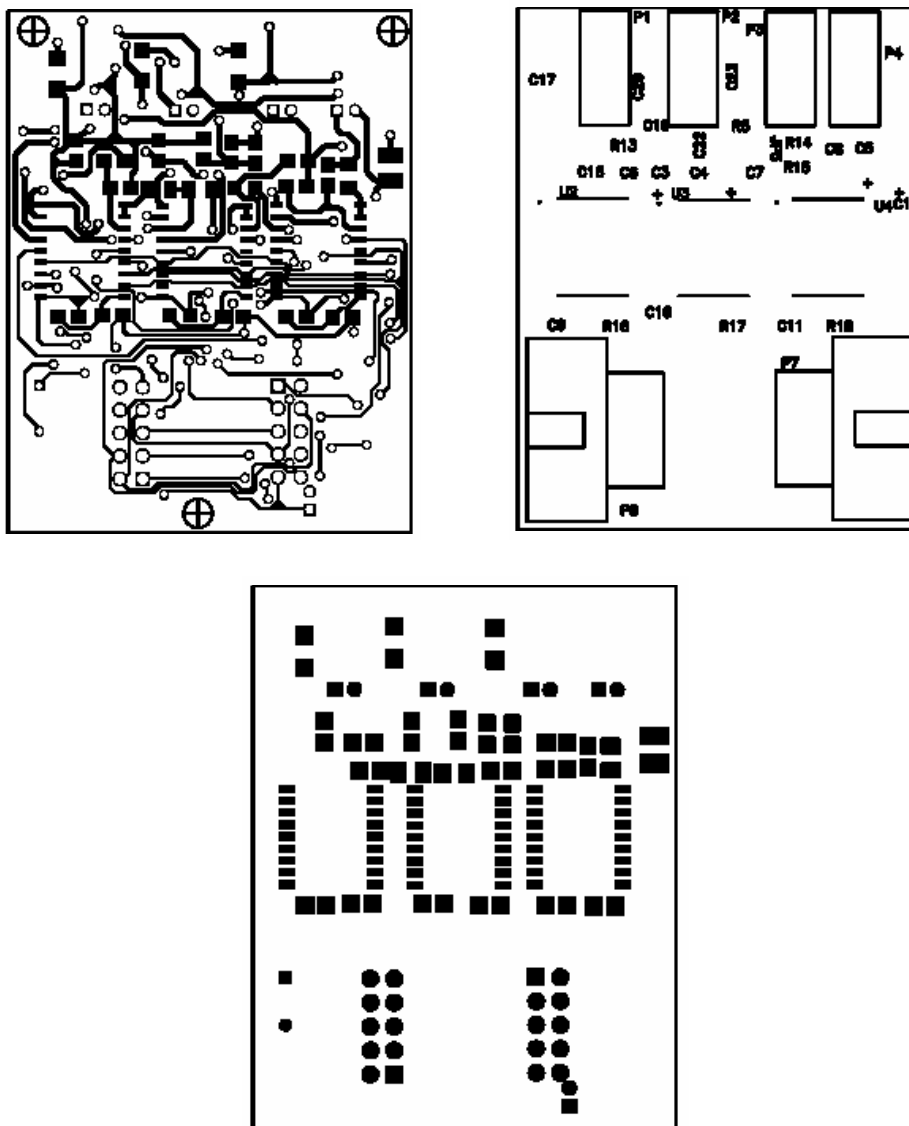
Figura A.2 – Face Superior
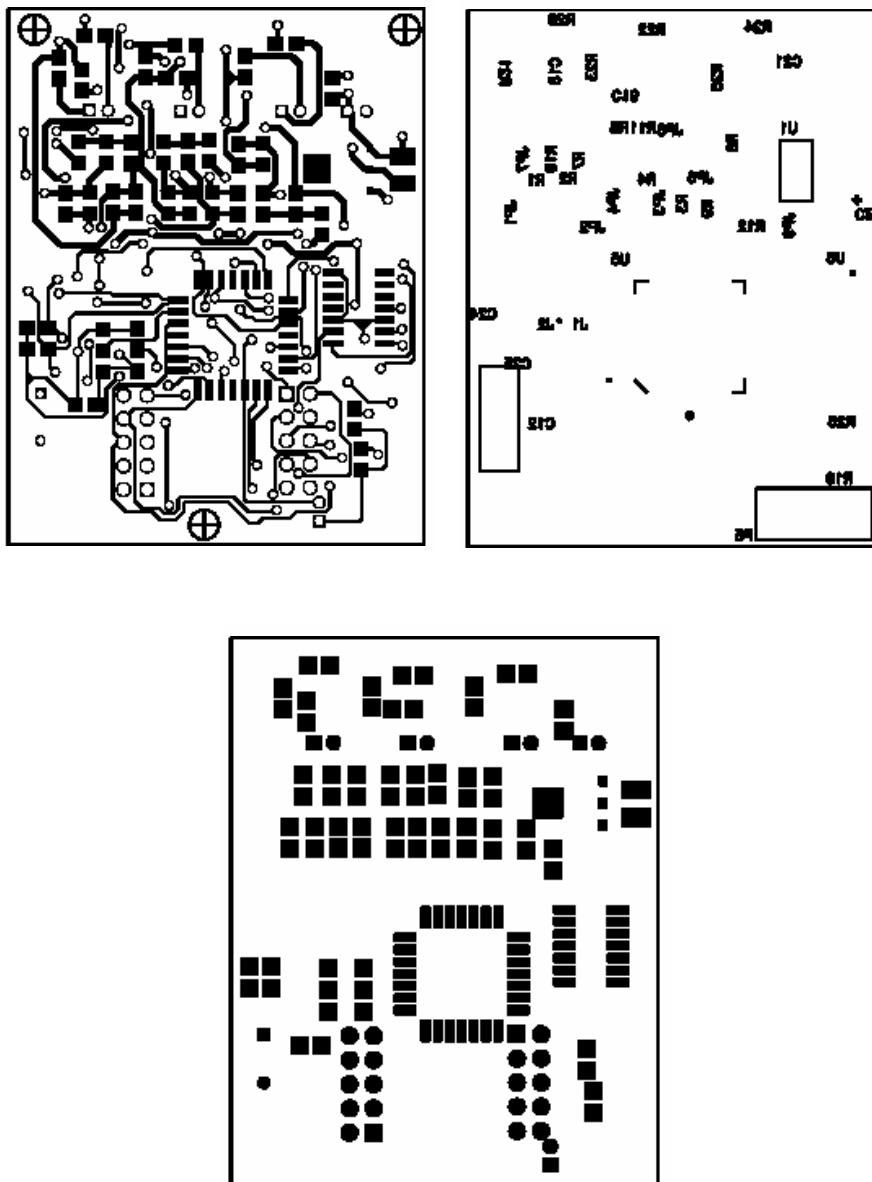
Figura A.3 – Face Inferior
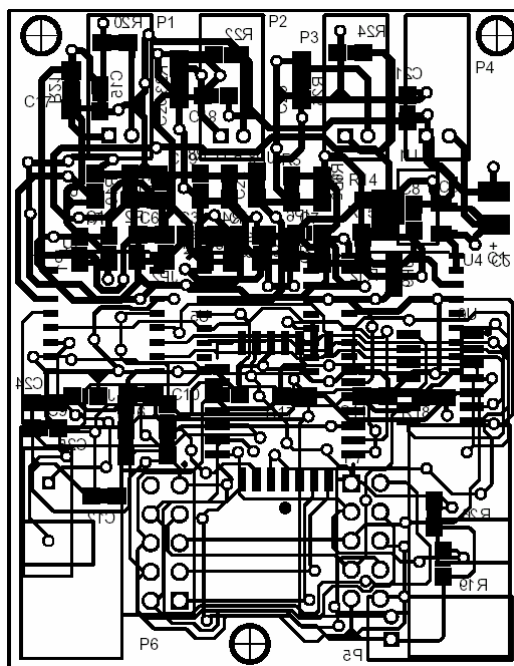
Figura A.4 – Placa Completa

A.4 Componentes

Os diversos componentes utilizados na construção da placa de circuito impresso são apresentados na tabela A.6.

Tipo	**Designação**	**Valor**	**Obs.**
Conversor A/D	**U2,U3,U4**	**ADS1210**	**18 Lead SOIC**
Regulador de Tensão	**U1**	**Reg 1117-5**	**SOT-223**
PAL	**U5**	**P5Z22V10**	**28 Pin PLCC**
Buffer	**U6**	MC74HC08AD	SOIC-14
Resistência	**R13,R14,R15,R16, R17,R18**	**10Ω**	**RC0805**
Resistência	**R7,R8,R9,R10, R11,R12**	**3KΩ**	**RC0805**
Resistência	**R1,R2,R3,R4,R5, R6,R19,R26**	**10KΩ**	**RC0805**
Resistência de Precisão	**R20,R21,R22,R23, R24,R25**	**10KΩ**	**RC0805**
Condensador	**C3,C4,C5**	**1µF**	**TAJ A**
Condensador	**C1,C2**	**10µF**	**TAJ B**
Condensador	**C6,C7,C8,C9,C10, C11,C12**	**0.1µF**	**CC0805**
Condensador de Precisão	**C17,C20,C23**	**0.1µF**	**CC1206**
Condensador	**C24,C25**	**22pF**	**CC0805**
Condensador	**C15,C16,C18,C19, C21,C22**	**1000pF**	**CC0805**
Cristal	**D1**	**9,8MHz**	
Ficha	**P1,P2,P3,P4,P5**	**CON2M79R**	
Ficha	**P6,P7**	**IDC10RM**	

Tabela A.6 – Lista de Material

B Manual do Driver

B.1 Introdução

Quando um hardware de aquisição de dados recebe um sinal analógico, converte-o numa tensão. Depois digitaliza-o usando um conversor analógico-digital, ficando deste modo, os dados, em situação de poderem ser entendidos por um computador. Para tal é necessário existir algo que possa controlar todas as operações de transferência de dados entre o utilizador e o hardware. Esta é a função da arquitectura de software, mais conhecida por driver.

B.2 Funções/Funcionalidades

As funções que se apresentam de seguida são aquelas que o utilizador tem ao seu dispor para controlar o hardware da maneira que pretender. As funções têm parâmetros de entrada que serve para indicar ao driver as configurações que se pretende.

Função DRIVER_CLOSE

A função serve para encerrar a comunicação entre o utilizador e o hardware.

A função chama-se da seguinte forma:

DRIVER_CLOSE()

Função DRIVER_OPEN

É a função principal do driver. É através dela que se inicia a comunicação entre o utilizador e o hardware. Além de inicializar a comunicação também, irá configurar o *comand register* dos ADS, seguindo-se de uma calibração e por fim irá efectuar uma sincronização entre todos os ADS activos.

Também serve para configurar um ou mais ADS e voltar a fazer os procedimentos já referidos.

A função aceita parâmetros de entrada e devolve um valor especifico de acordo com determinada acção. A função devolve sempre um valor do tipo *unsigned char*.

A função chama-se da seguinte forma:

DRIVER_OPEN(open_card,open_ads,gain,turbo mode,data rate,
data format , diff_mode,opc_calib,n_card,n_ads)

Parâmetros de entrada	Valores validos para os parâmetros de entrada	Significado
open_card	ALL_CARD	Inicia/configura a comunicação com todas as *cards* do sistema. Este parâmetro só é válido com a atribuição de ALL_ADS a open_ads.
	POS_CARD_1	Inicia/configura a comunicação com a *card* 1
	POS_CARD_2	Inicia/configura a comunicação com a *card* 2
	POS_CARD_3	Inicia/configura a comunicação com a *card* 3
	POS_CARD_4	Inicia/configura a comunicação com a *card* 4
open_ads	ALL_ADS	Inicia/configura a comunicação com todos os ADS do sistema. Este parâmetro só é válido com a atribuição de ALL_CARD a open_card.
	POS_ADS_1	Inicia/configura a comunicação com o ADS 1
	POS_ADS_2	Inicia/configura a comunicação com o ADS 2
	POS_ADS_3	Inicia/configura a comunicação com o ADS 3
Gain	GAIN_1	Ganho de amplificação do ADS de 1
	GAIN_2	Ganho de amplificação do ADS de 2
	GAIN_4	Ganho de amplificação do ADS de 4
	GAIN_8	Ganho de amplificação do ADS de 8
	GAIN_16	Ganho de amplificação do ADS de 16
turbo mode	TB_MODE_1	Atribui o valor de 1 ao *turbo mode*
	TB_MODE_2	Atribui o valor de 2 ao *turbo mode*
	TB_MODE_4	Atribui o valor de 4 ao *turbo mode*
	TB_MODE_8	Atribui o valor de 8 ao *turbo mode*
	TB_MODE_16	Atribui o valor de 16 ao *turbo mode*
data rate	de 10Hz a 1000Hz	O valor de *data rate* tem

		que estar neste intervalo
data format	DF_2_COMPLT	Indica que os dados de saída estão no formato de complemento para 2
	DF_OFFS_BIN	Indica que os dados de saída estão no formato de *offset binary*
diff mode	BIP_MODE	Indica que o sinal de entrada é simétrico.
	UNI_MODE	Indica que o sinal de entrada é unipolar.
opc_calib	DEFLT_CALIB_1	Efectua uma calibração por defeito, isto é, faz a *system offset calibration* seguido da *full-scale calibration* e por fim da *self calibration*.
	OFFST_CALIB_2	Efectua a *system offset calibration*.
	FULSC_CALIB_3	Efectua a *full-scale calibration.*
	SELF_CALIB_4	Efectua a *self-calibration*.
	PSDO_CALIB_5	Efectua a *pseudo system calibration.*
	BCKGD_CALIB_6	Efectua *a background calibration.*
n_card	MAX_CARD_1	Indica que existe 1 *card* no sistema.
	MAX_CARD_2	Indica que existe 2 *card* no sistema.
	MAX_CARD_3	Indica que existe 3 *card* no sistema.
	MAX_CARD_4	Indica que existe 4 *card* no sistema.
n_ads	MAX_ADS_1	Indica que existe 1 ADS por *card.*
	MAX_ADS_2	Indica que existe 2 ADS por *card*.
	MAX_ADS_3	Indica que existe 3 ADS por *card*.

Tabela B.1 – Parâmetros de entrada para Função DRIVER OPEN

Valor retomado pela função	Significado
OPEN_OK	A função foi executada sem problemas.
OPEN_NOT_OK_CFG_TBM_GAIN	Indica que o produto do ganho pelo *turbo mode* é superior a 16.
OPEN_NOT_OK_CFG_DEC_RTIO	Indica que o *decimation ratio* está fora do intervalo de resultados válidos, isto é, entre 19 e 8000.
OPEN_NOT_OK_CARD_OUT_LIM	O parâmetro de entrada para o open_card não está de acordo com a tabela anterior. A função não foi executada.
OPEN_NOT_OK_ADS_OUT_LIM	O parâmetro de entrada para o open_ads não está de acordo com a tabela anterior. A função não foi executada.
OPEN_NOT_OK_CFG_GAIN_OUT_LIM	O parâmetro de entrada para o gain não está de acordo com a tabela anterior. A função não foi executada.
OPEN_NOT_OK_CFG_TB_MODE_OUT_LIM	O parâmetro de entrada para o *turbo mode* não está de acordo com a tabela anterior. A função não foi executada.
OPEN_NOT_OK_CFG_D_RATE_OUT_LIM	O parâmetro de entrada para o *data rate* não está de acordo com o intervalo especificado na tabela anterior. A função não foi executada.
OPEN_NOT_OK_CFG_DF_OUT_LIM	O parâmetro de entrada para o *data format* não está de acordo com a tabela anterior. A função não foi executada.
OPEN_NOT_OK_CFG_MODE_OUT_LIM	O parâmetro de entrada para o diff_mode não está de acordo com a tabela anterior. A função não foi executada.
OPEN_NOT_OK_CALIB_OUT_LIM	O parâmetro de entrada para o opc_calib não está de acordo com a tabela anterior. A função não foi executada.
OPEN_NOT_OK_MAX_CARD_OUT_LIM	O parâmetro de entrada para o n_card não está de acordo com a tabela anterior. A função não foi executada.
OPEN_NOT_OK_MAX_ADS_OUT_LIM	O parâmetro de entrada para o n_ads não está de acordo com a tabela anterior. A função não foi executada.
OPEN_NOT_OK_SLEEP_ALL	Indica que todos os ADS estão em

	modo *sleep*, o que impossibilita uma nova configuração dos mesmos.
OPEN_NOT_OK_SLEEP	Indica que o ADS especificado pelo open_ads na card open_card se encontra no modo *sleep*, o que impossibilita uma nova configuração.
OPEN_NOT_OK_ALL	Indica que se usou o parâmetro ALL_CARD sem usar o parâmetro ALL_ADS, ou vice-versa.

Tabela B.2 – Variáveis de saída para Função DRIVER OPEN

Função ADS_READ

Esta função serve para ler os dados resultantes da conversão e que estão armazenados no DOR do ADS.

A função aceita parâmetros de entrada e devolve um valor especifico de acordo com determinada acção. A função devolve sempre um valor do tipo *unsigned long*.

A função chama-se da seguinte forma:

ADS_READ(ler_card,ler_ads)

Parâmetros de entrada	Valores validos para os parâmetros de entrada	Significado
ler_card	POS_CARD_1	Indica que o ADS que se pretende ler encontra-se na *card* 1.
	POS_CARD_2	Indica que o ADS que se pretende ler encontra-se na *card* 2.
	POS_CARD_3	Indica que o ADS que se pretende ler encontra-se na *card* 3.
	POS_CARD_4	Indica que o ADS que se pretende ler encontra-se na *card* 4.
ler_ads	POS_ADS_1	Indica que o ADS que se irá ler será o número 1 da *card* já referenciada.
	POS_ADS_2	Indica que o ADS que se irá ler será o número 2 da *card* já referenciada.
	POS_ADS_3	Indica que o ADS que se irá ler será o número 3 da *card* já referenciada.

Tabela B.3 – Parâmetros de entrada da Função ADS READ

Valor retomado pela função	Significado
READ_NOT_OK	Indica que a comunicação ainda não foi inicializada, logo impossível de ler os resultados da conversão.
READ_NOT_OK_CARD_OUT_LIM	O parâmetro de entrada para o ler_card não está de acordo com a tabela anterior. A função não foi executada.
READ_NOT_OK_ADS_OUT_LIM	O parâmetro de entrada para o ler_ads não está de acordo com a tabela anterior. A função não foi executada.
READ_NOT_OK_SLEEP_ALL	Indica que é impossivel ler o ADS especificado porque todos os ADS do sistema encontram-se em modo *sleep*.
READ_NOT_OK_SLEEP	Indica que é impossivel ler o ADS especificado por se encontrar em modo *sleep*. Deve-se primeiro tira-lo deste modo e então lê-lo.
VALOR DIFERENTE DOS ANTERIORES	Dá-nos o resultado da conversão.

Tabela B.4 – Variáveis de saída da Função ADS READ

Função DSYNC

A função serve para sincronizar todos os ADS que se encontram activos no sistema. Desta maneira existe uma sincronização entre todos ADS de modo que o DATA READY do ADS master seja igual a todos os outros, não se correndo o risco de haver a possibilidade de ler-se dados incorrectos, devido ao facto dos dados da conversão ainda não estarem em condições de serem lidos.

A função não aceita parâmetros de entrada e devolve um valor especifico de acordo com determinada acção. A função devolve sempre um valor do tipo *unsigned char.*

A função chama-se da seguinte forma:

DSYNC( )

Valor retomado pela função	Significado
DSYNC_NOT_OK	Indica que a comunicação ainda não foi inicializada, logo impossível de sincronizar os ADS.
DSYNC_OK	Indica que todos os ADS se encontram sincronizados com o ADS *master*.

Tabela B.5 – Variáveis de saída da Função DSYNC

Função CALIBRATE

A função serve para calibrar todos os ADS do sistema que se encontram activos. No fim de calibrar a função ainda sincroniza os ADS com o ADS *master*.

A função aceita parâmetros de entrada e devolve um valor especifico de acordo com determinada acção. A função devolve sempre um valor do tipo *unsigned char*.

A função chama-se da seguinte forma:

CALIBRATE(calib_card, calib_ads, opc_calib)

Parâmetros de entrada	Valores validos para os parâmetros de entrada	Significado
calib_card	ALL_CARD	Calibra todas as *cards* do sistema. É usado em conjunto com ALL_ADS.
	POS_CARD_1	Calibra a *card* 1
	POS_CARD_2	Calibra a *card* 2
	POS_CARD_3	Calibra a *card* 3
	POS_CARD_4	Calibra a *card* 4
calib_ads	ALL_ADS	Calibra todos os ADS. É usado em conjunto com ALL_CARD.
	POS_ADS_1	Calibra o ADS 1 da *card* especificada.
	POS_ADS_2	Calibra o ADS 2 da *card* especificada.
	POS_ADS_3	Calibra o ADS 3 da *card* especificada.
opc_calib	DEFLT_CALIB_1	Efectua uma calibração por defeito, isto é, faz a *system offset calibration* seguido da *full-scale calibration* e por fim da *self calibration*.
	OFFST_CALIB_2	Efectua a *system offset calibration.*
	FULSC_CALIB_3	Efectua a *full-scale calibration.*
	SELF_CALIB_4	Efectua a *self-calibration*.
	PSDO_CALIB_5	Efectua a *pseudo system calibration.*
	BCKGD_CALIB_6	Efectua a *background calibration.*

Tabela B.6 – Parâmetros de entrada da Função CALIBRATE

Valor retomado pela função	Significado
CALIB_NOT_OK	Indica que a comunicação ainda não foi inicializada, logo impossível de calibrar os ADS.
CALIB_NOT_OK_CARD_OUT_LIM	O parâmetro de entrada para o calib_card não está de acordo com a tabela anterior. A função não foi executada.
CALIB_NOT_OK_ADS_OUT_LIM	O parâmetro de entrada para o calib_ads não está de acordo com a tabela anterior. A função não foi executada.
CALIB_NOT_OK_CALIB_OUT_LIM	O parâmetro de entrada para o opc_calib não está de acordo com a tabela anterior. A função não foi executada.
CALIB_NOT_OK_SLEEP_ALL	Indica que é impossível calibrar o ADS especificado porque todos os ADS do sistema encontram-se em modo *sleep*.
CALIB_NOT_OK_SLEEP	Indica que é impossível calibrar o ADS especificado por se encontrar em modo *sleep*. Deve-se primeiro tira-lo deste modo e então calibra-lo.
CALIB_NOT_OK_ALL	Indica que se usou o parâmetro ALL_CARD sem o uso do parâmetro ALL_ADS.
CALIB_OK	Indica-nos que a calibração pretendida foi realizada com sucesso.

Tabela B.7 – Variáveis de Saída da Função CALIBRATE

Função ADS_SLEEP

A função serve para colocar os ADS em modo *sleep*.

É de notar que a função só permite colocar os ADS *masters* nesse modo, se colocar em modo *sleep* todos os outros ao mesmo tempo. Nos restantes casos os ADS *masters* nunca estarão em modo *sleep*.

A função aceita parâmetros de entrada e devolve um valor especifico de acordo com determinada acção. A função devolve sempre um valor do tipo *unsigned char*.

A função chama-se da seguinte forma:

ADS_SLEEP(sleep_card, sleep_ads)

Parâmetros de entrada	Valores validos para os parâmetros de entrada	Significado
sleep_card	ALL_CARD	Coloca em modo *sleep* todas as *cards* do sistema. É usado em conjunto com ALL_ADS.
	POS_CARD_1	Coloca em modo *sleep* a *card* 1
	POS_CARD_2	Coloca em modo *sleep* a *card* 2
	POS_CARD_3	Coloca em modo *sleep* a *card* 3
	POS_CARD_4	Coloca em modo *sleep* a *card* 4
sleep_ads	ALL_ADS	Coloca em modo *sleep* todos os ADS. È usado em conjunto com ALL_CARD.
	POS_ADS_1	Coloca em modo *sleep* o ADS 1 da *card* especificada.
	POS_ADS_2	Coloca em modo *sleep* o ADS 2 da *card* especificada.
	POS_ADS_3	Coloca em modo *sleep* o ADS 3 da *card* especificada.

Tabela B.8 – Parâmetros de entrada da Função ADS SLEEP

Valor retomado pela função	Significado
SLEEP_NOT_OK	Indica que a comunicação ainda não foi inicializada, logo impossível de colocar em modo *sleep* os ADS.
SLEEP_NOT_OK_CARD_OUT_LIM	O parâmetro de entrada para o sleep_card não está de acordo com a tabela anterior. A função não foi executada.
SLEEP_NOT_OK_ADS_OUT_LIM	O parâmetro de entrada para o sleep_ads não está de acordo com a tabela anterior. A função não foi executada.
SLEEP_NOT_OK_MST_CARD	Indica que é impossível colocar este ADS em modo *sleep* por ser o ADS *master*.
ALL_SLEEP_ALREADY	Indica que é impossível colocar em modo *sleep* o ADS especificado porque todos os ADS do sistema já se encontram nesse modo.
SLEEP_ALREADY	Indica que é impossível colocar em modo *sleep* o ADS especificado por já se encontrar nesse modo.

SLEEP_NOT_OK_ALL	Indica que se usou o parâmetro ALL_CARD sem o uso do parâmetro ALL_ADS.
SLEEP_OK	Indica-nos que a tarefa pretendida foi realizada com sucesso.

Tabela B.9 – Variáveis de saída da Função ADS SLEEP

Função ADS_WAKEUP

A função serve para tirar os ADS do modo *sleep*, isto é colocá-los em modo *wakeup*.

É de referir que se os ADS estão todos em modo *sleep* então só existe uma maneira de os tirar desse modo, que é colocar em modo *wakeup* todos os ADS ao mesmo tempo. Este facto deve-se aos ADS *masters* terem que estar no modo *wakeup* sempre que um ADS qualquer também esteja nesse modo.

Após a colocação dos ADS em modo *wakeup*, procede-se a uma calibração por defeito e de uma sincronização. A calibração por defeito é uma *system offset calibration* seguida de uma *full-scale calibration* e por último de uma *self-calibration*.

A função aceita parâmetros de entrada e devolve um valor especifico de acordo com determinada acção. A função devolve sempre um valor do tipo *unsigned char*.

A função chama-se da seguinte forma:

ADS_WAKEUP(wakeup_card, wakeup_ads)

Parâmetros de entrada	Valores validos para os parâmetros de entrada	Significado
Wakeup_card	ALL_CARD	Coloca em modo *wakeup* todas as *cards* do sistema. É usado em conjunto com ALL_ADS.
	POS_CARD_1	Coloca em modo *wakeup* a *card* 1
	POS_CARD_2	Coloca em modo *wakeup* a *card* 2
	POS_CARD_3	Coloca em modo *wakeup* a *card* 3
	POS_CARD_4	Coloca em modo *wakeup* a *card* 4
Wakeup_ads	ALL_ADS	Coloca em modo *wakeup* todos os ADS. É usado em conjunto com ALL_CARD.
	POS_ADS_1	Coloca em modo *wakeup* o ADS 1 da *card* especificada.
	POS_ADS_2	Coloca em modo *wakeup* o ADS 2 da *card* especificada.

	POS_ADS_3	Coloca em modo *wakeup* o ADS 3 da *card* especificada.

Tabela B.10 – Parâmetros de entrada da Função ADS WAKEUP

Valor retomado pela função	Significado
WAKEUP_NOT_OK	Indica que a comunicação ainda não foi inicializada, logo impossível de colocar em modo *wakeup* os ADS.
WAKEUP_NOT_OK_CARD_OUT_LIM	O parâmetro de entrada para o wakeup_card não está de acordo com a tabela anterior. A função não foi executada.
WAKEUP_NOT_OK_ADS_OUT_LIM	O parâmetro de entrada para o wakeup_ads não está de acordo com a tabela anterior. A função não foi executada.
WAKEUP_NOT_OK_ALL_SLEEP	Indica que é impossível colocar este ADS em modo *wakeup* por o ADS *master* ainda se encontrar no modo *sleep*.
ALL_WAKEUP_ALREADY	Indica que é impossível colocar em modo *wakeup* o ADS especificado porque todos os ADS do sistema já se encontram nesse modo.
WAKEUP_ALREADY	Indica que é impossível colocar em modo *wakeup* o ADS especificado por já se encontrar nesse modo.
WAKEUP_NOT_OK_ALL	Indica que se usou o parâmetro ALL_CARD sem o uso do parâmetro ALL_ADS.
WAKEUP_OK	Indica-nos que a tarefa pretendida foi realizada com sucesso.

Tabela B.11 – Variáveis de saída da Função ADS WAKEUP

C Circuito Programável (P5Z22V10)

C.1 Características Gerais

O circuito utilizado para gerar os sinais de CS, SDOUT, DRDY e DSYNC é a P5Z22V10 da *Philips*.

Características especificas

- 5V de tensão de alimentação;
- Tecnologia CMOS;
- Baixo consumo de potência e elevada rapidez;
- Permite 1000 escritas/apagamentos;
- Reprogramável usando o hardware adequado;
- Capacidade de provocar um *reset* assíncrono;
- Sinal de relógio comum a todos os *flip-flops*;

Na figura seguinte encontra-se a configuração dos pinos do *package* utilizado (28-PLCC), seguida de uma tabela com a descrição dos mesmos.

Figura C.1 – Configuração dos pinos

Nome do pino	Descrição
I1-I11	Entradas dedicadas
NC	Desligado
F0-F9	Entradas/Saídas
I0/CLK	Entrada dedicada/Entrada de relógio
Vcc	Tensão de alimentação
GND	Massa

Tabela C.1 – Descrição dos Pinos

Como se pode ver na tabela existem 12 pinos que funcionam apenas como entradas, e 10 que podem funcionar como entradas ou saídas. Caso se queira utilizar o relógio dos *flip-flops* internos, este terá de ser ligado ao pino 2 (I0/CLK), e é o mesmo para todos os *flip-flops* do circuito (ver diagrama lógico).

Na figura seguinte encontra-se o diagrama lógico de um dispositivo deste tipo, onde se pode observar a matriz de ligação, os *flip-flops* e os respectivos sinais de controlo e a forma como as saídas são implementadas.
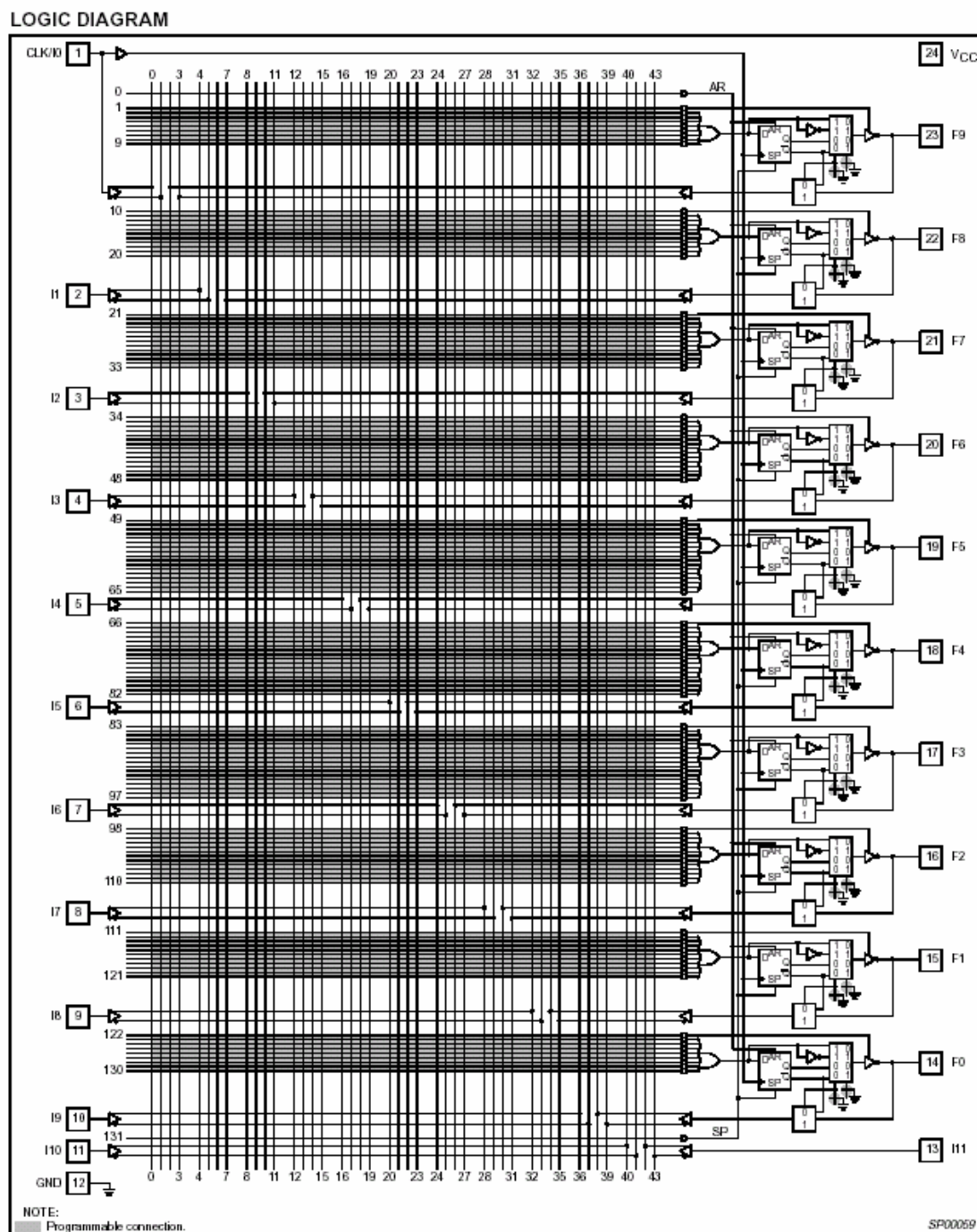
Figura C.2 – Diagrama Lógico

C.2 Listagem da programação utilizada

```
MODULE      pal_tfc
TITLE 'programa da pal para tfc'

DECLARATIONS


"INPUTS
ADDR0            PIN    4;
ADDR1            PIN    9;
ADDR2            PIN    7;
ADDR3            PIN    6;
SDOUTAD0         PIN    18;
SDOUTAD1         PIN    17;
SDOUTAD2         PIN    16;
ADDR3AUX         PIN    10;
ADDR2AUX         PIN    3;
CLK              PIN    2;
XIN              PIN    11;
IN_DRDY0         PIN    25;
IN_DRDY          PIN    5;
IN_SDOUT         PIN    13;

"OUTPUTS

DRDY             PIN    26;
SDOUT            PIN    27;
CSAD0            PIN    23;
CSAD1            PIN    21;
CSAD2            PIN    20;
DSYNC            PIN    24;
XIN_OUT          PIN    19;


EQUATIONS

CSAD0=((ADDR3!$ADDR3AUX)&(ADDR2!$ADDR2AUX)&!ADDR1&!ADDR0)!
$(ADDR1&ADDR0)#(!ADDR3&!ADDR2&ADDR1&ADDR0);

CSAD1=((ADDR3!$ADDR3AUX)&(ADDR2!$ADDR2AUX)&!ADDR1&ADDR0)!$
(ADDR1&ADDR0)#(!ADDR3&!ADDR2&ADDR1&ADDR0);

CSAD2=((ADDR3!$ADDR3AUX)&(ADDR2!$ADDR2AUX)&ADDR1&!ADDR0)!$
(ADDR1&ADDR0)#(!ADDR3&!ADDR2&ADDR1&ADDR0);

SDOUT=(!CSAD0&SDOUTAD0#!CSAD1&SDOUTAD1#!CSAD2&SDOUTAD2)#(
CSAD0&CSAD1&CSAD2&IN_SDOUT);

DRDY=(IN_DRDY # (IN_DRDY0 & (!CSAD0 # !CSAD1 # !CSAD2) ) );
```

```
DSYNC.clk=CLK;
DSYNC:=!(!ADDR3&ADDR2&ADDR1&ADDR0);

XIN_OUT=!XIN;

END
```

Referências Bibliográficas

DAVID A. JOHNS & KEN MARTIN: “Analog Integrated Circuit Design”, JOHN WILEY & SONS, 1997. ISBN 0-471-14448-7
SEITZER, D., PRETZI, G., HAMDY: “Electronic analog-to-digital converters”, JOHN WILEY & SONS, 1993. ISBN 0-471-90198-9
[1] Artigo “High-Performance Mixed-Signal Design”

Alguns sites de pesquisa:

http://www.prosig.com/signal-processing/
http://www.datatranslation.com
http://www.precisionresistor.com
http://www.burrbrown.com